

EP 29837 (1)

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2001年10月25日 (25.10.2001)

PCT

(10) 国際公開番号
WO 01/80431 A1

(51) 国際特許分類: H03M 7/30, G06F 19/00

(21) 国際出願番号: PCT/JP01/03324

(22) 国際出願日: 2001年4月18日 (18.04.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2000-117343 2000年4月19日 (19.04.2000) JP
特願2000-149122 2000年5月19日 (19.05.2000) JP

(71) 出願人 および

(72) 発明者: 大森 聡 (OMORI, Satoshi) [JP/JP]; 〒338-0832 埼玉県さいたま市西堀四丁目11番7号627 Saitama (JP).

DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:
— 国際調査報告書

(81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM,

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: NUCLEOTIDE SEQUENCE INFORMATION, AND METHOD AND DEVICE FOR RECORDING INFORMATION ON SEQUENCE OF AMINO ACID

(54) 発明の名称: ヌクレオチドの配列情報及びアミノ酸の配列情報を記録するための方法及び装置

STANDARD SAMPLE E					
j	i	1 標準法行E	2	3 A(i, j)	4 C(j)
1	1	1b1c3ed2	d82560cd	eeddd410	0011ded3
2	2	1861b42d	7cada747	03c031cf	4bc5e2c0
3	3	cbf0a0cc	58c66212	13003c84	72208e8d
4	4	0263c628	a3ca28a3	8a3ca217	097656d2
5	5	67214088	4001a98a	d21495bf	11f192805
6	6	c24c70ba	38471d07	e5972387	38008427
7	7	fed9df69	33ed4060	da161561	75a29ebb
8	8	b6aa6803	8a0a2b5d	54d3d000	a3c65a14
9	9	db12a0ce	1936909c	1fda421f	4952e69a
10	10	6a1a55fa	9b5983d0	0b19e41c	4316a030
11	11	08de6535	4f11df55	18765318	e9b666d3
12	12	f69d6440	37939a3c	d69673c4	19997882
13	13	7cb7ce4e	979002db	587558f2	b90eca79
14	14	33db47a2	a69cf658	1a63e9bd	388d76d3
15	15	5851e29a	5c340059	0b5d7615	26097e92
16	16	cb6bd976	d6dadfc9	9a4f7d91	1132527d
A(1, 17)					
17	17	255eccad	92a6785d	a9364517	d07937ac
18	18	a1419351	blacbe59	b01f6e2a	a82812aa
19	19	39a87e84	ca16b410	022940eb	18a1725a
20	20	e3d76869	d341243c	a5e0563f	a0ed0c23
21	21	5837e19f	cd7a5534	054d7963	596667bf
22	22	61937899	a9cfe9d7	6d3c9838	e1a43218
23	23	e1fe951a	20192ddd	91b42560	d85047ec
24	24	ad42d010	5bc651a6	d61d2509	6d6813b9
25	25	75c5d35c	d98a1675	4ee5903e	1da62d6a
26	26	66833823	de88f6e1	53bed09b	83bb79d7
27	27	030934d9	2bb59d99	12c384db	7e0ca4e1
28	28	7ce41d1d	d3d67975	9159766d	b5182d06
29	29	78601b5b	10c08cce	4bc9d6d9	77da0b90
30	30	5bb6e283	7235af0e	d402d614	10b5981a
31	31	01ja7f0e	e566f019	ae740433	8edb42a5
32	32	eada4be2	1a1a161d	65343363	21ad9905
33	33	215ab0c2	1d08e241	1831400c	71919005
B2(i)	B2(i)	aa1a481c	e448c00e	3558b731	43d91669

(57) Abstract: A method and device for recording information on the sequence of a nucleotide in a nucleic acid or gene or information on the sequence of an amino acid in a protein in an irreducible minimum amount of data. The mathematical abstract value of text data expressing the sequence of a nucleotide of one line is calculated to confirm the identity of two sequences by using the abstract value. The text data is converted into binary data according to a predetermined conversion table. The binary data is divided into conversion data A(i,j) of a plurality of lines and a plurality of columns. Each column of the conversion data A(i,j) is calculated in the direction of sequence to determine a syndrome C(j) (j=1, 2, ...). Each row of the conversion data A(i,j) is calculated in a direction different from the direction of sequence to determine two syndromes B1(i), B2(i) (i=1, 2, ...). The sequence of a nucleotide can be approximated by the syndromes C(j), B1(i), B2(i).

[続葉有]

WO 01/80431 A1



(57) 要約:

核酸や遺伝子中のヌクレオチドの配列情報、又はタンパク質中のアミノ酸の配列情報をできるだけ少ないデータ量で記録するための方法及び装置である。一列のヌクレオチドの配列を示すテキストデータの数学的な要約値を算出し、この要約値を用いて2つの配列の同一性等を確認する。更に、そのテキストデータを所定の変換テーブルに従ってバイナリーデータに変換し、このバイナリーデータを複数行で複数列の変換データ $A(i, j)$ に分割する。各列の変換データ $A(i, j)$ を配列方向に演算してシンδροーム $C(j)$ ($j = 1, 2, \dots$) を求め、各行の変換データ $A(i, j)$ を非配列方向に演算して2組のシンδροーム $B1(i)$, $B2(i)$ ($i = 1, 2, \dots$) を求め、シンδροーム $C(j)$, $B1(i)$, $B2(i)$ によってヌクレオチドの配列を近似的に表す。

明 細 書

ヌクレオチドの配列情報及びアミノ酸の配列情報を記録するための方法及び装置

5

技術分野

本発明は、例えばDNA（デオキシリボ核酸：deoxyribonucleic acid）又はRNA（リボ核酸：ribonucleic acid）等の核酸や遺伝子の少なくとも一部を構成する一群のヌクレオチドの配列情報、及びタンパク質の少なくとも一部を構成する一群のアミノ酸の配列情報の記録方法及び装置に関する。更に本発明は、その配列情報を供給するためのビジネスモデルとして好適な配列情報の供給方法、その配列情報を記録したコンピュータ読み取り可能な記録媒体、及びその配列情報の記録方法を使用する場合に好適な要約値算出方法に関する。

15

背景技術

人間、及び他の生物（動物、植物、微生物等）のDNAを構成する1対のヌクレオチドの鎖（又は塩基の鎖）の配列情報の解読が世界的に行われている。この場合、従来よりDNAを構成する4種類のヌクレオチドは、塩基としてアデニンを含むヌクレオチド、グアニンを含むヌクレオチド、シトシンを含むヌクレオチド、及びチミンを含むヌクレオチドにそれぞれ文字A、G、C、及びTを割り当てることによって、それぞれ1バイト（＝8ビット）のテキストデータで表わされている。その結果として一つのDNAの配列は、それを構成する1対の重合体の鎖の内の一方の鎖のヌクレオチド（n個とする）の配列を順次文字A、G、C、T（又はa、g、c、t）の何れかで表すことによって、nバイトのテキストデータで表されていた。同様に、一つのRNAを構成する

20

25

1本の n 個のヌクレオチドの配列は、チミンを含むヌクレオチドの代わりにウ
ラシルを含むヌクレオチドに文字U（又はu）を割り当てることによって、 n
バイトのテキストデータで表されていた。

これに関して、例えば人間の最も大きい第1染色体中のDNAの配列は、約
5 2億5千万個のヌクレオチドの配列であり、最も小さい第22染色体中のDN
Aの配列は、約5000万個のヌクレオチドの配列であるため、人間の各染色
体中のDNAの配列は、約250Mバイト～50Mバイトのテキストデータで
表すことができる。更に、一人の人間の全部のDNA情報（ゲノム）は、約3
0億個のヌクレオチドの配列で表すことができるため、そのゲノムは、約3G
10 バイトのテキストデータで記録することができる。なお、それらのテキストデ
ータに対して通常のファイル圧縮技術を適用することによって、それらのテキ
ストデータは、例えば元のデータの50%程度の圧縮ファイルとしても記録、
又は送信することができる。

また、DNAの配列の解読に続いて、DNA中の多数の遺伝子の情報に基づ
15 いてそれぞれ合成されるタンパク質の機能の研究も広く行われている。この場
合、タンパク質を構成する20種類のアミノ酸は、三文字表記（3-Letter Cod
e）ではそれぞれ3文字（例えばAla, Cys, Glu等）のテキストデー
タで表され、一文字表記（1-Letter Code）ではそれぞれ1文字のテキストデー
タ（例えばA, C, E等）で表されるため、 n 個のアミノ酸よりなるタンパ
20 ク質の配列は、 n バイトのテキストデータで表すことができる。そして、種々
のタンパク質は、それらのアミノ酸が約20個～約1000個程度所定の順序
で配列されたものであるため、それらのタンパク質の配列は、最大でも約1k
バイト程度のテキストデータで記録することができる。また、例えば人間の遺
伝子の総数は約3万個と予想されており、それに対してタンパク質は理論的な
25 ものも含めて約10万種類の存在が可能であると言われている。

上記の如く例えば一人の人間のDNA情報をテキストデータで記録するため

には、全部で3 Gバイト程度の記憶容量が必要であり、仮に通常の圧縮ファイルの技術を適用しても1 Gバイト程度の記憶容量が必要である。また、人間以外の大腸菌や各種ウィルス等のDNA情報も解析されて次第に公開されるようになっているが、これらのDNA情報をテキストデータの形で多く集めると、
5 数100 Mバイト程度の記憶容量が必要である。これはRNAの配列情報についても同様である。

このように人間又は他の生物のDNA情報をテキストデータ、又はこの通常の圧縮ファイルの形で記録するものとする、例えば1枚の記憶容量が5 Gバイト程度のDVD-ROM (digital video disc-ROM) ディスクのように膨大な
10 記憶容量を持つ記録媒体が必要である。更に、そのDNA情報を利用する場合にその記録媒体からの読み出し時間が長くなり、処理時間が長くなるという不都合がある。

また、現状の一般の通信回線の通信速度は、最大で1 M b p s 程度であるため、例えば1 Gバイト程度のDNA情報をその通信回線を介して送信するものとすると、送信時間は最短でも約2時間程度となり、あまり実用的ではない。
15 特に最近はそのDNA情報をデジタルの携帯電話システムを介して送信する場合も考えられるが、現在の携帯電話システムの通信速度はせいぜい100 k b p s 程度であるため、少なくとも人間のDNA情報の伝送で使用することは困難である。

次に、例えば或る微生物のDNA中の遺伝子について複数の研究者が並行して研究するような場合に、複数の研究者が保有している標準となるDNA（又は遺伝子）のヌクレオチドの配列の同一性をどのように保証するのかという問題がある。即ち、そのDNAのヌクレオチドの配列が例えば数Mバイト（文字
20 数で数100万文字）程度のテキストデータで記録されている場合に、複数の
25 研究者が互いに自分のテキストデータと他人のテキストデータとの同一性（完全一致性）を短時間に確認するのは必ずしも容易ではない。

これに関連して、例えば人間又は他の生物のDNA情報の利用方法としては、標準的なDNAの配列と、検査対象のDNAの配列との間の相違する部分をサーチする場合が考えられる。これは、いわゆるSNP（一塩基変位多型：Single Nucleotide Polymorphism）の可能性を検査するような場合に必要になると考えられる。しかしながら、両方のDNAのヌクレオチドの配列がそれぞれ膨大なテキストデータで表わされている場合に、それら2つのテキストデータを比較して相違点を検出するにはかなりの長い時間が必要となり、検査時間が長くなるという不都合がある。

更に、人間又は他の生物のDNA情報を製薬会社の研究者等のユーザに提供するビジネスも行われつつあるが、この場合に、複数の情報供給者間で重複した情報の提供をできるだけ避けることが望ましい。このためには、複数の情報供給者間で、DNAのヌクレオチドの全体の配列情報を公開することなく、ヌクレオチドの配列の実質的な同一性を容易に確認できるようにすることが望ましい。更に、情報供給者が例えば通信回線を介してDNA情報をユーザに提供する場合には、できるだけ少ない情報量で、即ち短い送信時間で必要な情報をユーザに提供できるビジネスモデルが必要である。また、ユーザ側では、提供されたDNA情報に伝送エラー等が無いかどうかを容易に確認できることが望ましい。上記の各課題はRNAや遺伝子のヌクレオチドの配列情報についても同様に当てはまるものである。

更に、一つのタンパク質のアミノ酸の配列は、最大でも約1kバイト程度のテキストデータで記録することができるが、タンパク質の種類は理論的に約10万個程度にもなるため、全部のタンパク質の配列情報をテキストデータで表すと、全部のDNAの配列情報程度の膨大な量となる。従って、個々のタンパク質の配列は、できるだけ少ない情報量で記録できることが望ましい。また、2つのタンパク質の配列情報の同一性を容易に確認できるシステムも必要である。

本発明は斯かる点に鑑み、核酸や遺伝子中の一系列のヌクレオチドの配列情報、及びタンパク質中の一系列のアミノ酸の配列情報をできるだけ少ないデータ量で記録できる記録方法及び記録装置を提供することを第1の目的とする。

5 また、本発明は、2つのヌクレオチドの配列情報同士、又は2つのアミノ酸の配列情報同士の同一性を少ないデータ量で高精度に確認できる記録方法及び記録装置を提供することを第2の目的とする。

更に本発明は、2つのヌクレオチドの配列情報の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元できる記録方法及び記録装置を提供することを第3の目的とする。

10 また、本発明は、一系列のヌクレオチドの配列情報、又は一系列のアミノ酸の配列情報を少ないデータ量でユーザに提供できるビジネスモデルを提供することを第4の目的とする。

更に本発明は、そのビジネスモデルにおいて、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性、又は相違する部分を少ないデータ量で容易に確認できるようにすることをも目的とする。

15 また、本発明は、ヌクレオチドの配列情報が少ないデータ量で記録されたコンピュータ読み取り可能な記録媒体を提供することをも目的とする。

また、本発明は、ヌクレオチド又はアミノ酸の配列情報を記録する場合に使用できる効率的な要約値の計算方法を提供することを目的とする。

20

発明の開示

本発明によるヌクレオチドの配列情報の記録方法は、一系列のヌクレオチドの配列情報の記録方法であって、その一系列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一系列のヌクレオチドの配列に関する情報を記録するものである。

25

斯かる本発明によれば、その一系列のヌクレオチドは、例えばDNA (deoxyr

ibonucleic acid) を構成する 1 対の重合体の鎖の一方の鎖の少なくとも一部、RNA (ribonucleic acid) を構成する 1 列の重合体の鎖の少なくとも一部、又は遺伝子の構成を表す一列のヌクレオチドの配列の少なくとも一部である。そして、その一列のヌクレオチドの配列は、各ヌクレオチドに含まれる塩基の配列ともみなすことができる。本発明によれば、そのヌクレオチドの配列が、そのテキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、DVD-ROM のような大容量の媒体の他に、CD-ROM やフラッシュ ROM のような小容量でも通常のコンピュータで手軽に再生できる媒体を使用できる。

更に、少ないデータ量の配列情報であれば、通信回線を介して短時間に送信できるため、実質的に安価に配列情報の供給を行うことが可能となる。

本発明において、その一列のヌクレオチドは 4 種類のヌクレオチドよりなり、その 4 種類のヌクレオチドを互いに異なる 6 ビット以下のデータで表すことが望ましい。テキストデータ形式では、各ヌクレオチドは、それぞれ 8 ビットのアスキーコード (ASCII CODE) 、即ち文字 A, G, C, T (又は U) の何れかで表されるため、各ヌクレオチドを 6 ビット以下のデータで表すことによって、データ量を減らすことができる。

なお、テキストデータが記録されたファイルが通常の圧縮技術 (ZIP ファイル、LHA ファイル等) で圧縮できるように、本発明のデータが記録されたファイルも更に通常の圧縮技術で圧縮して記録できることは言うまでもない。但し、圧縮されたファイルを使用する場合には、解凍作業が必要になり、最終的には元のファイルを復元する必要があるため、元のファイル自体のデータ量を減らしておくことは極めて有効である。

また、その 4 種類のヌクレオチドを互いに異なる 2 ビットのデータで表すことが望ましい。2 ビットのデータによって、最も少ないデータ量で 4 種類のヌクレオチド (又は塩基) を表すことができる。

また、その一列のヌクレオチドが、一つのDNAを構成する1対の重合体の鎖の内の1本の鎖の全部又は一部であるときに、その4種類のヌクレオチド中の互いに相補的な2対のヌクレオチドをそれぞれ互いにビット反転の関係にある1対のデータで表すことが望ましい。互いに相補的な2対のヌクレオチドとは、互いに相補的な2対の塩基と実質的に同じ意味である。ここで、2進数で表現した数 k を $\text{bin}(k)$ として、例えば図2のDNA(5)に示すように、アデニンを含むヌクレオチド(7A)を $\text{bin}(00)$ で表したとき、それに対して相補的なチミンを含むヌクレオチド(7T)を $\text{bin}(11)$ で表す。更に、グアニンを含むヌクレオチド(7G)を $\text{bin}(01)$ で表したとき、それに対して相補的なシトシンを含むヌクレオチド(7C)を $\text{bin}(10)$ で表す。この結果、DNA(5)の一方のヌクレオチドの鎖(6A)が $\text{bin}(0001101111\cdots)$ (=BNAとする)で表されて、それと相補的な他方のヌクレオチドの鎖(6B)に対応する2進数のデータBNBは、コンピュータによって2進数BNAをビット毎に反転するだけで極めて高速に求めることができる。

次に、本発明において、より具体的な第1の記録方法は、その一列のヌクレオチドの配列に関する情報を、その配列を表すテキストデータ又は数値データの数学的な要約値(message digest)で表すものである。この数学的な要約値は、暗号理論において、送信ファイルの作成者の本人確認を行うために、送信ファイルに所定のハッシュ関数を施すことによって得られる値と数学的には同等のものである。しかしながら、本発明においては、一列のヌクレオチドの配列を表すデータ(原データ)の要約値を、例えば最先の解読者の主張や、2つの膨大な原データの同一性の確認に使用する点が本質的に異なっている。即ち、或るDNAのヌクレオチドの配列を最初に解読した者が、その配列を示す原データの要約値を例えばインターネット上で公開することによって、原データを公開することなく最先に解読したことを主張できる。また、例えば情報供給者からDNAの配列情報を購入したユーザは、購入した配列情報の要約値を、例

例えばインターネット上で公開されているそのDNAの要約値と比較することによって、購入した配列情報の同一性を高い確率で確認できる。更に、複数の研究者が同一のDNAについて研究を行う場合に、各研究者が保持しているDNAのヌクレオチドの膨大な配列情報の長さ、及び要約値を求め、これらと比較
5 することによって、研究対象の同一性を容易に高い確率で確認することができる。

この場合、その一列のヌクレオチドが25個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報を40ビット以上で192ビット以下の長さの数学的な要約値で表すことが望ましい。25個以上の
10 ヌクレオチドの配列のテキストデータは、200ビット（＝25・8ビット）以上になるため、その要約値を192ビット以下とすることで、テキストデータよりも少ないデータ量となる。また、特に処理単位が64ビットのコンピュータを使用する場合には、要約値の長さは64ビットの倍数、即ち64ビット、128ビット、又は192ビットが望ましいと考えられる。

また、例えば将来的に全人類のDNAの配列情報を必要に応じて解読するような状況を想定して、世界人口を100億人程度と仮定すると、そのDNAの配列情報は約 10^{10} 通りにもなる。更に、安全係数を100倍程度とすると、その要約値は、 10^{12} （＝ $10^{10} \cdot 100$ ）通り、即ち約 $2^{39.86}$ 通り以上の値を取る必要がある。このためには、その要約値を40ビット以上の長さとする
20 ればよい。これによって、2つのDNA又はRNAの配列情報同士の同一性を 10^{-12} 以上の精度で高精度に確認できる。

更に、その数学的な要約値は、その一列のヌクレオチドの配列に対応するテキストデータ又は数値データにMD5ハッシュ関数、又はSHS（Secure Hash Standard）ハッシュ関数の演算を施して得ることができる。この場合、MD
25 5ハッシュ関数は、高速演算が可能であると共に、得られる要約値が128ビットであり、通常のコンピュータで処理し易い利点がある。一方、SHSハッ

シュ関数は、元のデータの推定がより困難であるが、得られる要約値が160ビットと、通常のDNA又はRNAのヌクレオチドの配列の表現に関しては必要以上に長いと考えられる。従って、通常のヌクレオチドの配列の表現については、MD5ハッシュ関数がより実用的と考えられる。

5 また、暗号理論で使用されるハッシュ関数は、送信ファイルの内容が推定されないように、かつ内容の衝突の確率が極めて低くなるように設計されるため、その要約値は例えば最低でも128ビット程度の長さが必要とされると共に、複雑な演算が繰り返して実行される。これに対して本発明で使用するハッシュ関数は、通常の互いに異なるヌクレオチドの配列に対してほぼ衝突が無ければ
10 よいため、あまり複雑な演算を繰り返して行う必要は無いと考えられる。但し、通常の暗号理論で要約値の演算対象となるファイルは、せいぜい1Mバイト程度の長さであるのに対して、本発明で使用するハッシュ関数の演算対象は、例えば人間のDNAのヌクレオチドの配列とすると、100Mバイト程度にも達する膨大なデータのファイルである。そこで、本発明で使用するハッシュ関数
15 （ハッシュ演算プログラム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

次に、本発明において、より具体的な第2の記録方法は、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数
20 行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータ $T(i, j)$ に分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データ $A(i, j)$ に変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム（syndrome）情
25 報 $B1(i)$ 、 $B2(i)$ を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報 $C(j)$ を求

め、その第1組及び第2組のシンδροーム情報でその一列のヌクレオチドの配列を表すものである。

本発明においては、テキストデータを複数行で複数列の部分テキストデータに分割した後に、各部分テキストデータをそれぞれ変換データに変換しているが、これは予めそのテキストデータを一列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えば図7に示す部分テキストデータ $T(i, j)$ を集めたテキストデータの情報の大部分を、例えば図9に示す第1組のシンδροーム情報 $B1(i)$ 、 $B2(i)$ 、及び第2組のシンδροーム情報 $C(j)$ で表すことができる。具体的に、図7のテキストデータを配列方向に N 個($i=1 \sim N$)で、非配列方向に M 個($j=1 \sim M$)の部分テキストデータ $T(i, j)$ に分割し、各部分テキストデータ $T(i, j)$ が16個分のヌクレオチドのテキストデータを含むものとする、元のテキストデータのデータ量 $DT1$ は、以下のようになる。

$$DT1 = 16 \cdot N \cdot M \text{ (バイト)} \quad \dots (1)$$

更に、各ヌクレオチドを2ビットのデータで表すものとする、各部分テキストデータ $T(i, j)$ は、それぞれ32ビットの変換データ $A(i, j)$ に変換され、シンδροーム情報 $B1(i)$ 、 $B2(i)$ 、 $C(j)$ も32ビットのデータとなる。また、非配列方向のシンδροーム情報が2列 $B1(i)$ 、 $B2(i)$ あるとすると、シンδροーム情報のデータ量 $DS1$ は、以下のようになる。

$$\begin{aligned} DS1 &= 32(2 \cdot N + M) \text{ (ビット)} \\ &= 4(2 \cdot N + M) \text{ (バイト)} \quad \dots (2) \end{aligned}$$

従って、仮に $N=64$ 、 $M=128$ とすると、(1)式及び(2)式よりデータ量 $DT1$ 、 $DS1$ は以下のようになる。

$$DT1 = 131072 \text{ (バイト)} \approx 130 \text{ kバイト} \quad \dots (3)$$

$$DS1 = 1024 \text{ (バイト)} = DT1 / 128 \quad \dots (4)$$

従って、シンδροーム情報のデータ量は、元のテキストデータのデータ量の
 ほぼ $1/100$ 程度に圧縮できる。この場合、例えば人間の1本の染色体のD
 NAの配列は、50Mバイト～250Mバイト程度のテキストデータで表され
 5 るため、予めそのテキストデータを500個～2500個程度のブロックに分
 割し、各ブロック毎にシンδροーム情報を求めることによって、全部のシンド
 ローム情報のデータ量はそのテキストデータのほぼ $1/100$ 程度、即ち50
 0kバイト～2.5Mバイト程度に圧縮される。この程度のデータ量であれば、
 例えば携帯電話システムのような低速の通信回線を介しても短時間に送信でき
 10 ると共に、DVD-ROMよりも容量の少ないCD-ROM等の記録媒体にも
 余裕を持って記録することができる。

この場合、複数行のその変換データの各行の変換データをそれぞれその非配
 列方向に交互に第1群の変換データ（例えば奇数番目の変換データ $A(i, 1)$,
 $A(i, 3)$, \dots) 及び第2群の変換データ（例えば偶数番目の変換デ
 ータ $A(i, 2)$, $A(i, 4)$, \dots) に分けたとき、その第1の演算は、所
 15 定の整数Kを用いてその第1群の変換データ、及びその第2群の変換データの
 それぞれの法Kのものと和を求める演算であり、その第2の演算は、複数列の
 その変換データの各列の変換データに対する法Kのものと和を求める演算であ
 る。その変換データ $A(i, j)$ をsビット（例えば $s = 32$, $s = 64$ 等）
 20 とすると、その整数Kは一例として次のようになる。

$$K = 2^s \quad \dots (5)$$

通常のコンピュータでは、その法Kのものと和演算は極めて高速に実行する
 ことができる。

また、その一列のヌクレオチドの配列を基準配列として、この基準配列の2
 25 組のそのシンδροーム情報 ($B1(i)$, $B2(i)$, $C(j)$) に対応させ
 て、検査対象の一列のヌクレオチドの配列 ($TF(i, j)$) の2組のシンド

ローム情報 ($B1F(i)$, $B2F(i)$, $CF(j)$) を求め、その4組のシンドローム情報よりその基準配列に対するその検査対象の一行のヌクレオチドの配列の相違部を求めることが望ましい。例えば図7の配列を基準配列、図10の配列を検査対象の配列として、図7の基準配列のシンドローム情報が図8に、図10の配列のシンドローム情報が図11に表されている。このとき、図8のシンドローム情報 ($B1(i)$, $B2(i)$, $C(j)$) に対して、図11のシンドローム情報 ($B1F(i)$, $B2F(i)$, $CF(j)$) は、 $B1F(1)$, $B2F(4)$, $CF(16)$, $CF(17)$ の値が異なるため、それらの交点として、図10の部分テキストデータ $TF(4, 16)$, $TF(1, 17)$ が図7の基準配列と異なっていることを検出できる。即ち、4組のシンドローム情報を比較することによって、少ないデータ量の比較で、検査対象の配列のどの部分テキストデータが基準配列と異なっているかを検出できる。

この際に、基準配列と異なっている部分をエラーコード (error code) と呼ぶと、エラーコードが部分テキストデータの各行、又は各列に一つである場合には、それら4組のシンドローム情報、及びその検査対象のエラーコードに対応する変換データの法Kの加減算より、基準配列の変換データ $A(4, 16)$, $A(1, 17)$ 、ひいては部分テキストデータ $T(4, 16)$, $T(1, 17)$ が正確に復元できる。従って、例えば遺伝子中の一つの塩基 (ヌクレオチド) だけが異なる SNP (一塩基変位多型: Single Nucleotide Polymorphism) は本発明によって容易に検出できると共に、それに対応する正常な配列も容易に復元できる。

なお、図10の場合のように隣接する2つの列の部分テキストデータ $TF(4, 16)$, $TF(1, 17)$ に跨るような長いエラーコード (以下、「バーストエラー (burst error)」と呼ぶ) が存在する場合に、非配列方向のシンドロームが各行に1つ (即ち、 $B1F(i)$ と $B2F(i)$ との和) のみであ

るとすると、1行中の2箇所の部分テキストデータ、及び1列中の2箇所の部分テキストデータにエラーコードが検出されてしまう。従って、エラーコードの位置の誤検出が生じて、それに対応する基準配列の復元も困難となる。これに対して本発明のように各行で2つのシンδροーム情報を求めることによって、

5 バーストエラーの検出及び復元を正確に行うことができる。なお、各行で2つのシンδροーム情報を求める代わりに、配列方向（各列）で例えば前半分と後半分との2群の変換データに対して2つのシンδροーム情報を求めるようにしてもよく、どちらを採用するかは全体のデータ量が少なくなるように選択すればよい。

10 次に、本発明の記録装置は、一列のヌクレオチドの配列情報の記録装置であって、一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置（4）と、この配列読み取り装置で読み取られた配列の情報をテキストデータとして第1ファイル（19）に記録する第1記録手段（ステップ102～104）と、その第1ファイルのテキストデータより

15 も少ないデータ量で、その配列読み取り装置で読み取られた配列の情報を表し、この配列の情報を第2ファイル（20，21）に記録する第2記録手段（ステップ105～107）とを有するものである。これによって、本発明の配列情報の記録方法が実施できる。

この場合、その第2記録手段は、一例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列を、この配列を表すテキストデータ又は数値データの数学的な要約値で表すものである。

また、その第2記録手段は、別の例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てる

25

ことによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報を求め、その第1組及び第2組のシンδροーム情報をその第2ファイルに記録するものである。

また、本発明の記録媒体は、一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、その一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一列のヌクレオチドの配列に関する情報が記録されたものである。本発明によれば、例えば人間のDNA又は遺伝子のヌクレオチドの配列情報を、少ないデータ量で記録できるため、記録媒体としてCD-ROM、CD-R、フラッシュROM等の使い勝手の良い媒体を使用できる。また、記録媒体としてDVD-ROMやハードディスク装置等の大容量の記録媒体を使用した場合には、極めて多くの試料のヌクレオチドの配列情報を記録することができる。

この場合、その一列のヌクレオチドが25個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報は、一例として40ビット以上で192ビット以下の長さの数学的な要約値でその記録媒体に記録されるものである。この場合には、記録媒体としてフレキシブルディスクであっても使用できる。

また、別の例として、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2

の演算を施して第2組のシンドローム情報を求めておき、その一列のヌクレオチドの配列に関する情報は、その第1組及び第2組のシンドローム情報としてその記録媒体に記録される。この記録媒体を用いることによって、例えば2つの試料のヌクレオチドの配列の相違する部分の位置の検出ができると共に、その相違する部分が少ない場合にはそれに対応する配列の復元を行うことができる。

次に、本発明の配列情報の供給方法は、一列のヌクレオチドの配列情報の供給方法であって、その一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者（2A）が、その一列のヌクレオチドの配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線（1）を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧したユーザ（2B）より、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

この供給方法は、上記の本発明のヌクレオチドの配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る生物XのDNAのヌクレオチドの配列を最初に解読した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にその生物XのDNAの配列を解読したことを主張できる。更に、ユーザが同じ配列情報を異

なる供給者から誤って購入することも防止できる。

また、或るユーザが、その供給者よりその生物XのDNAの配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

この場合、その一列のヌクレオチドは25個以上のヌクレオチドの配列であるときに、一例として、その数学的な要約値は、40ビット以上で192ビット以下のデータであり、その供給者は、更にその一列のヌクレオチドの所定の一部の配列の情報をその通信回線を介して閲覧可能な状態にしておくことが望ましい。その要約値、及びその配列の長さの他に、そのように例えばその配列の先頭の8個程度、及び後端の8個程度の配列を比較することによって、同一性の確認をより高精度に行うことができる。

また、その供給者は、その一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応するその数値データを第1ファイル(19)に記録して保持し、その供給者は、そのテキストデータ、又はその数値データを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分データに分割し、その部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報を求め、その第1組及び第2組のシンδροーム情報を第2ファイル(20)に記録して保持し、第1段階としてそのユーザは、その供給者よりその第2ファイルに記録されている2組のシンδροーム情報を受け取り、その2組のシンδροーム情報に基づいて検査対象の一列のヌクレオチドの配列の内の

その供給者の一列のヌクレオチドの配列との相違部を特定し、この相違部の配列の復元ができない場合に、第2段階としてそのユーザはその供給者よりその第1ファイルに記録されているそのテキストデータ、又はその数値データの内のその配列の復元ができない部分の情報の提供を要求することが望ましい。

5 このようにそのユーザが最初は、希望するヌクレオチドの配列情報のシンドローム情報のみを購入する場合には、そのデータ量の小さいシンドローム情報はその通信回線を介して短時間で受信することができる。そして、シンドローム情報だけで検査対象の配列のエラーコードの特定、及び復元ができる場合には、それ以上の配列情報を購入する必要が無い。一方、エラーコードが多く存在し、シンドローム情報のみでは全部の正確なデータが復元できない場合には、
10 復元できない部分のテキストデータのみを購入することによって、通信回線を介して必要な配列情報を短時間に購入できる。従って、通信回線として、携帯電話システムのような比較的低速の通信回線も使用できる。

次に、本発明のアミノ酸の配列情報の記録方法は、一列のアミノ酸の配列情報
15 の記録方法であって、その一列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、その一列のアミノ酸の配列に関する情報を記録するものである。

斯かる本発明によれば、その一列のアミノ酸は、例えば或るタンパク質を構成するアミノ酸の配列の少なくとも一部である。そのアミノ酸の配列が、その
20 テキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、小容量でも通常のコンピュータで手軽に再生できる媒体を使用できると共に、通信回線を介して送信する際の時間を短縮できる。

本発明において、その一列のアミノ酸は、一つのタンパク質を構成する1本のアミノ酸の鎖の全部又は一部である場合に、一例としてその一列のアミノ酸
25 の配列に対応するテキストデータが、20種類のアミノ酸に対して互いに異なる6ビット以下のデータを割り当てることによって変換される。テキストデー

タ形式で一文字表記 (1-Letter Code) を行うものとする、20種類のアミノ酸は、それぞれ8ビットのアスキーコード (ASCII CODE)、文字では例えば A, C, E等で表されるため、各アミノ酸を6ビット以下のデータで表すことによって、データ量を減らすことができる。

5 なお、一つのアミノ酸の種類は一系列の3個のヌクレオチドの配列、即ち一つの遺伝子コドン (codon) によって決定される。これに関して、上記のヌクレオチドの配列情報の記録方法において、各ヌクレオチドを2ビットのデータで表した場合に、1つの遺伝子コドンは6ビットのデータで表される。そこで、この各遺伝子コドンの6ビットのデータを、対応するアミノ酸の6ビットのデータとみなしてもよい。この場合には、所定のアミノ酸を表すデータが複数存在する、即ちコードの縮重 (degeneracy) が生じるため、一例として各アミノ酸のデータの内で最も小さいデータをそのアミノ酸に割り当てるようにしてもよい。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。また、20種類のアミノ酸は、最も少ないデータ量では、5ビット
10 のデータで表すことができる。

 また、本発明において、より具体的な第1の記録方法は、その一系列のアミノ酸の配列に関する情報を、その配列を表すテキストデータの数学的な要約値 (message digest) で表すものである。例えば所定のハッシュ関数を用いてそのテキストデータの要約値を求め、この要約値をインターネット上で公開することによって、そのテキストデータを公開することなく、その配列を最先に解読したことを主張 (証明) できる。更にそのテキストデータを購入したユーザが、購入したデータの要約値を求め、この要約値を公開されている要約値と比較することによって、購入したデータの同一性を確認できる。
20

 この場合、その一系列のアミノ酸は25個以上のアミノ酸の配列であるときに、
25 その一系列のアミノ酸の配列に関する情報を16ビット以上で192ビット以下の長さの数学的な要約値で表すことが望ましい。タンパク質の種類は、仮想的

なものも含めて10万（＝10⁵）種類程度と言われており、次の関係が成立している。

$$10^5 \approx 2^{16.6} \quad \dots (6)$$

従って、例えばアミノ酸の配列の個数も識別データに用いるものとする、
5 16ビット以上の要約値を用いることによって、ほぼ全てのタンパク質を識別
することができる。また、25個以上のアミノ酸の配列のテキストデータは、
一文字表記でも200ビット以上になるため、192ビット以下の要約値のデ
ータ量はテキストデータのデータ量よりも少なくなる。

また、その数学的な要約値は、その一列のアミノ酸の配列に対応するテキス
10 トデータに例えばMD5ハッシュ関数（要約値は128ビット）、又はSHS
（Secure Hash Standard）ハッシュ関数（要約値は160ビット）の演算を施
して得られる。この場合、要約値が必要以上に長くない点ではMD5ハッ
シュ関数が望ましい。但し、タンパク質を構成するアミノ酸の配列の数は20
個～1000個程度であり、要約値から元のテキストデータが推定される恐れ
15 がある。そこで、アミノ酸の配列の要約値を算出する場合で、かつ元のテキス
トデータの秘匿性を高めたい場合には、より複雑な演算を行って得られる要約
値も長くなるSHSハッシュ関数を使用することが望ましい。

また、本発明において、より具体的な第2の記録方法は、その一列のアミノ
酸の配列に対応するテキストデータを、そのアミノ酸の配列方向に複数行で、
20 かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割
し、その部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに
異なる8ビット以下の数値データを割り当てることによって変換データに変換
し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して
第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎
25 にその配列方向に第2の演算を施して第2組のシンδροーム情報を求め、その
第1組及び第2組のシンδροーム情報でその一列のアミノ酸の配列を表すもの

である。

本発明は、予めそのテキストデータ（一文字表記とする）を一系列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えばアミノ酸の配列に対応するテキストデータを配列方向にN個（ $i = 1 \sim N$ ）で、非配列方向にM個（ $j = 1 \sim M$ ）の部分テキストデータ $T(i, j)$ に分割し、各部分テキストデータ $T(i, j)$ が4個分のアミノ酸のテキストデータを含むものとする、元のテキストデータのデータ量 $DT2$ は、以下ようになる。

$$DT2 = 4 \cdot N \cdot M \text{ (バイト)} \quad \dots (7)$$

更に、その部分テキストデータ $T(i, j)$ をそのまま変換データ $A(i, j)$ とみなすと、変換データ $A(i, j)$ はそれぞれ32ビットの数値データとなり、シンδροーム情報も32ビットのデータとなる。また、非配列方向のシンδροーム情報が2列あるとすると、シンδροーム情報のデータ量 $DS2$ は、以下ようになる。

$$\begin{aligned} DS2 &= 32 (2 \cdot N + M) \text{ (ビット)} \\ &= 4 (2 \cdot N + M) \text{ (バイト)} \quad \dots (8) \end{aligned}$$

従って、仮に $N = 16$, $M = 16$ とすると、(7) 式及び (8) 式よりデータ量 $DT2$, $DS2$ は以下ようになる。

$$DT2 = 1024 \text{ (バイト)} \quad \dots (9)$$

$$DS2 = 192 \text{ (バイト)} \doteq DT2 / 5.3 \quad \dots (10)$$

従って、シンδροーム情報のデータ量は、元のテキストデータのデータ量のほぼ $1/5$ 程度に圧縮できる。個々のタンパク質の配列のテキストデータのデータ量は1kバイト程度以下であるが、例えば10000種類程度のタンパク質のテキストデータをまとめると10Mバイト程度になる。この際にシンδροーム情報を用いることによって、通信回線を介して短時間で近似的な情報を送信することができる。また、シンδροーム情報を比較することによって、標準

試料のアミノ酸の配列と検査対象のアミノ酸の配列との相違部（エラーコード）を効率的に検出することができる。更に、エラーコードが各行、又は各列に1つの変換データのみであるときには、それに対応する正確な配列を復元できる。

特に、複数行のその変換データの各行の変換データをそれぞれその非配列方向に交互に第1群の変換データ及び第2群の変換データに分けたとき、その第1の演算は、所定の整数Kを用いてその第1群の変換データ、及びその第2群の変換データのそれぞれの法Kのものと和を求める演算であり、その第2の演算は、複数列のその変換データの各列の変換データに対する法Kのものと和を求める演算である場合には、2列に跨るような長い配列の相違（バーストエラー）であっても正確に検出、及び復元を行うことができる。

次に、本発明の一系列のアミノ酸の配列情報の記録装置は、一つのタンパク質の少なくとも一部に含まれる一系列のアミノ酸の配列情報をテキストデータとして第1ファイルに記録する第1記録手段と、その第1ファイルのテキストデータよりも少ないデータ量で、その一系列のアミノ酸の配列の情報を表し、この配列の情報を第2ファイルに記録する第2記録手段とを有するものである。この発明によって、本発明のアミノ酸の配列情報の記録方法を実施することができる。

この場合、その第2記録手段は、その一系列のアミノ酸の配列を、この配列を表すテキストデータの数学的な要約値で表すことが望ましい。

また、本発明の一系列のアミノ酸の配列情報の供給方法は、その一系列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる8ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者が、その一系列のアミノ酸の配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧し

たユーザより、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

5 この供給方法は、上記の本発明のアミノ酸の配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る新規のタンパク質のアミノ酸の配列を最初に決定した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にそのタンパク質の配列を決定したことを主張（証明）できる。更に、ユーザが同じ配列情報を異なる供給者から誤って購入することも防止でき、競合メーカは重複投資を避けることができる。

15 また、或るユーザが、その供給者よりそのタンパク質の配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

20 この場合、その一列のアミノ酸が25個以上のアミノ酸の配列であるときに、その数学的な要約値は、16ビット以上で192ビット以下のデータであることが望ましい。

 次に、本発明の第1の要約値の計算方法は、一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、その一つ又は複数のファイルに記録されたデータの内で所定のコードを無視して要約値を計算するものである。

本発明によれば、例えばヌクレオチドの配列を表すテキストデータの要約値を計算する場合に、その配列を見やすくするために所々にスペース、改行、及びそれまでのヌクレオチドの数を示す数字等が付加されていても、これらの付加されたコードを無視することによって、本来のヌクレオチドの配列に対応する要約値を計算することができる。

この場合、その無視する所定のコードの別の例は、同一又は互いに異なる2組のコード、及びこれら2組のコードに挟まれたデータである。即ち、例えばいわゆるコメント文を要約値の計算対象から除去することによって、コメント文の内容を任意に記載しても、本来のヌクレオチドの配列に対応する要約値を計算することができる。

この場合、そのテキストデータは、その所定のコードの他に25個以上のヌクレオチドの配列に関するデータを含むものとする。一例として、その要約値は、40ビット以上で192ビット以下のデジタルデータである。また、その要約値を計算するための関数としては、MD5ハッシュ関数（要約値は128ビット）、又はSHAハッシュ関数（要約値は160ビット）等を使用することができる。

また、本発明において、その一つ又は複数のファイルから1文字分のコードデータを読み出す毎に、この読み出されたコードデータがその所定のコードであるときには、この読み出されたコードデータを無視して、次の1文字分のコードデータの読み出しを行い、この読み出しによって得られたその所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うようにしてもよい。

このように部分的に読み出されたデータ毎にその所定のコードを無視して順次要約値の計算を行うことによって、例えば最初にその一つ又は複数のファイルからその所定のコードを取り出した新たなファイルを作成して、この新たなファイルの要約値を計算する方法と比べて、記憶装置の容量がほぼ1/2程度

で済む利点がある。

次に、本発明の第2の要約値の計算方法は、一連のテキストデータの要約値を計算するための要約値の計算方法であって、その一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、
5 その所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、その複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、この複数のファイルに記録されたテキストデータからその分割の順序に従って順次要約値を計算するものである。

10 斯かる本発明によれば、例えば人間のゲノム情報のような膨大な量のテキストデータの要約値を求める場合に、そのテキストデータを複数のファイルに分割して記録しておき、分割されたファイルのデータに順次演算処理を施すことができる。従って、CD-ROMやフレキシブルディスクのように比較的容量の少ない記録媒体を用いる場合にも、その膨大な量のテキストデータの要約値
15 を容易に、かつ正確に計算できる。

この場合、そのその所定個数ずつのコードデータ、及びその所定個数よりも少ない個数のコードデータからは、所定のコードデータ（例えば数字コード、スペースコード、改行コード等）を除外してもよい。

また、その所定個数は、要約値を計算する際のデータ量の単位に応じて定めることが望ましい。例えばMD5ハッシュ関数は、512ビット（64バイト）のデータ単位で要約値を計算するため、一つのコードデータが8ビット（1バイト）であるとする、その所定個数は、64の整数倍に設定することによって、各部分テキストデータ毎の要約値の計算が容易になる。

25 図面の簡単な説明

図1は、本発明の実施の形態の一例で使用されるコンピュータシステムを示

す概略構成図である。図2は、その実施の形態の一例で処理対象とするDNA、及びそのヌクレオチドの配列のバイナリーデータによる表現の例を示す図である。図3は、その実施の形態の一例におけるDNA情報の供給者の動作の一部を示すフローチャートである。図4は、図3の動作に続くDNA情報の供給者の動作を示すフローチャートである。図5は、その実施の形態の一例におけるDNA情報のユーザの動作の一部を示すフローチャートである。図6は、図5の動作に続くDNA情報のユーザの動作を示すフローチャートである。図7は、標準試料E（DNA）のヌクレオチドの配列（2048個）を表すテキストデータを4行で32列の部分テキストデータ $T(i, j)$ に分割した状態を示す図である。図8は、標準試料Eの変換データ $A(i, j)$ 、及びこれらから算出されるシンδροーム $C(j)$ 、 $B1(i)$ 、 $B2(i)$ を示す図である。図9は、標準試料Eのシンδροーム $C(j)$ 、 $B1(i)$ 、 $B2(i)$ を示す図である。図10は、試料F（DNA）のヌクレオチドの配列（2048個）を表すテキストデータを4行で32列の部分テキストデータ $TF(i, j)$ に分割した状態を示す図である。図11は、試料Fの変換データ $AF(i, j)$ 、及びこれらから算出されるシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ を示す図である。図12は、試料Fのシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ 、及び復元された変換データを示す図である。図13は、試料G（タンパク質）のアミノ酸（820個）の配列を表すテキストデータを8行で26列の部分テキストデータに分割した状態を示す図である。図14は、図13中の一部のデータを示す図である。図15は、本発明の実施の形態の第1の要約値計算シーケンスを示すフローチャートである。図16は、本発明の実施の形態の第2の要約値計算シーケンスを示すフローチャートである。図17は、本発明の実施の形態の表示画面内でのカーソルの移動方法の一例を示す図である。

発明を実施するための最良の形態

以下、本発明の好ましい実施の形態の一例につき図面を参照して説明する。
本例は、所定のDNA（デオキシリボ核酸：deoxyribonucleic acid）のヌク
レオチドの配列情報をコンピュータシステムで処理する場合に、本発明を適用
5 したものである。

図1は、本例のコンピュータシステム2Aの概略構成を示し、この図1にお
いて、コンピュータシステム2Aの中心は、CPU（中央演算処理ユニット）、
RAM、ROM等のメモリ、及びハードディスク装置等の記憶装置等からなる
情報処理装置10である。情報処理装置10には、ビデオRAM（VRAM）
11を介してCRTディスプレイよりなる表示装置12が接続されると共に、
10 I/Oユニット（入出力装置）14を介して、記録可能なCD-Recordableデ
ィスク（以下、「CD-R」と言う）16に対するデータの書き込み、及びC
D-RやCD-ROMからのデータの読み込みを行うことができるCD-R/
RWドライブ15が接続されている。情報処理装置10には、I/Oユニット
15 14を介して更に大容量の記憶装置としての記憶容量が数10Gバイト程度の
磁気ディスク装置17が接続されている。

本例の情報処理装置10中のハードディスク装置には、予めCD-R/RW
ドライブ15を介してオペレーティングシステム、及び後述のようにDNAの
配列情報を処理するためのアプリケーション・プログラムがインストールされ
20 ている。また、CD-R16が本発明の記録媒体に対応しているが、記録媒体
としては、CD-RやCD-ROMの他に、フラッシュROM、フレキシブル
ディスク、光磁気ディスク（MO）、デジタルビデオディスク（DVD）、又
はハードディスク装置（例えばインターネットを介して接続できるサーバに備
えられたもの）等を使用することができる。

25 情報処理装置10には更に、文字情報の入力装置としてのキーボード13、
ポインティング・デバイス（入力装置）としての光学式のマウス204、及び

ルータ（又はモデム等でもよい）よりなる通信制御ユニット 18 が接続されている。マウス 204 は、表示装置 12 の表示画面上のカーソルの位置を指定する信号を発生する変位信号発生部 207、選択すべき情報を指定する信号や各種コマンド等を発生するための左スイッチ 204 a 及び右スイッチ 204 b
5 （信号発生装置）を備えている。情報処理装置 10、VRAM 11、表示装置 12、キーボード 13、マウス 204、I/O ユニット 14、CD-R/RW ドライブ 15、磁気ディスク装置 17、及び通信制御ユニット 18 等よりコンピュータシステム 2A が構成されている。オペレーティングシステムとして本例では Windows（Microsoft Corporation の登録商標）を使用している。
10 なお、オペレーティングシステムとして、それ以外の UNIX（X/Open の登録商標）、OS/2（IBM Corporation の登録商標）、Mac OS（Apple Computer の登録商標）、又は Linux（Linus Torvalds の商標又は登録商標）等を使用する場合にも本発明が適用できることは言うまでもない。

そして、コンピュータシステム 2A（情報処理装置 10）は、通信制御ユニット 18 を介して一般電話回線よりなる通信ネットワーク 1 に接続され、通信
15 ネットワーク 1 には各種コンテンツのプロバイダ 3、及び別のコンピュータシステム 2B、及び不図示の多くのサーバやコンピュータシステムが接続されている。また、本例のコンピュータシステム 2A、2B 及びプロバイダ 3 は、通信ネットワーク 1 を介するインターネットによって相互に接続されている。こ
20 の場合、コンピュータシステム 2A の所有者が DNA 情報の供給者（販売者）であり、コンピュータシステム 2B の所有者がその DNA 情報のユーザ（購入者）である。そして、後者のコンピュータシステム 2B には、予め前者のコンピュータシステム 2A と同様の DNA の配列情報を処理するためのアプリケーション・プログラムがインストールされている。

25 さて、本例のコンピュータシステム 2A の情報処理装置 10 には、I/O ユニット 14 を介して DNA 中の一列のヌクレオチドの配列（又は塩基の配列）

を読み取るための配列読み取り装置としてのシーケンサー (DNA Sequencer) 4
が接続されている。シーケンサー 4 は、一例としてサンガーの方法 (Sanger m
ethod) によって DNA を構成する 1 対の重合体の鎖の一方の鎖のヌクレオチド
の配列を読み取る。サンガーの方法は、例えば文献 1 (Maxim D. Frank-Kamen
5 etskii: Unraveling DNA (the most important molecule of life, revised a
nd updated), translated by Lev Liapin, Chapter 6 (pp. 59-70) (Perseus Book
s, 1997)) に開示されている。シーケンサー 4 は、読み取った一列のヌクレオ
チドの配列をテキストデータ形式で内部の大容量の記憶装置に記憶すると共に、
情報処理装置 10 からの要求に応じて、その記憶装置中の所定のヌクレオチド
10 の配列のテキストデータを I/O ユニット 14 を介して情報処理装置 10 に供
給する。これに対して情報処理装置 10 は、DNA の配列情報を処理するた
めのアプリケーション・プログラムに基づいて以下の処理を行う。なお、シー
ケンサー 4 の代わりに、DNA 及び RNA (リボ核酸: ribonucleic acid) 等の
核酸を構成する一列のヌクレオチドの配列 (又は塩基の配列) の情報のデータ
15 ベースを接続してもよい。

先ず、本例の情報処理装置 10 の第 1 の基本的な処理動作につき説明する。
情報処理装置 10 は、シーケンサー 4 から供給される所定の DNA のヌクレオ
チドの配列を示すテキストデータを磁気ディスク装置 17 中のマスターファイ
ル 19 にそのまま記録すると共に、そのテキストデータをよりデータ量の少な
20 い数値データに変換し、この変換後の数値データを磁気ディスク装置 17 中の
ワーキングファイル 20 に記録する。なお、以下の説明において、2 進数表示
の数 k は $\text{bin}(k)$ で、16 進数表示の数 k は $\text{hex}(k)$ で表すものとする。

この場合、DNA は 4 種類のヌクレオチドより構成されており、シーケン
サー 4 から供給されるテキストデータ中では、塩基としてアデニン (adenine)
25 を含むヌクレオチド、グアニン (guanine) を含むヌクレオチド、シトシン (c
ytosine) を含むヌクレオチド、及びチミン (thymine) を含むヌクレオチドが

それぞれ文字A, G, C, 及びTで表されている。そして、文字A, G, C, 及びTには、データ上ではそれぞれhex (41), hex (47), hex (43), hex (54) よりなる1バイト (8ビット) のアスキーデータが割り当てられている。また、RNAの場合には、チミンを含むヌクレオチドの代わりにウラシル (uracil) を含むヌクレオチドが、文字U (hex (55)) で表されている。従って、n個のヌクレオチドの配列を示すテキストデータのデータ量はnバイトとなる。なお、それらのn個のヌクレオチドの配列は、n個の塩基 (アデニン、グアニン、シトシン、チミン (又はウラシル)) の配列ともみなすことができる。

本例ではそのテキストデータを、情報量を少なくすることなく最も少ないデータ量で表すために、DNA中の4種類のヌクレオチドを互いに異なる2ビットのデータで表す。この際に、DNAにおいては、1対の塩基 (アデニン及びチミン) が互いに相補的であり、別の1対の塩基 (グアニン及びシトシン) が互いに相補的である。そこで、相補的な塩基を含む1対のヌクレオチドを互いに相補的であるとして、1対の互いに相補的なヌクレオチド、即ちアデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、互いにビット反転の関係にある1対のデータを割り当て、別の1対の互いに相補的なヌクレオチド、即ちグアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに、互いにビット反転の関係にある別の1対のデータを割り当てる。本例ではそのデータの割り当てとして表1 (変換テーブル) を用いる。なお、表1は、ヌクレオチドの配列を示すテキストデータ中の文字A, T (又はU), G, C, をそれぞれbin (00), bin (11), bin (01), bin (10) で置換することを意味している。

《表1》

ヌクレオチド	2ビットのデータ
アデニンを含むヌクレオチド (A)	bin (00)
チミン (ウラシル) を含むヌクレオチド (T又はU)	bin (11)
グアニンを含むヌクレオチド (G)	bin (01)

シトシンを含むヌクレオチド (C) bin(10) 。

なお、本例では各ヌクレオチドを2ビットのデータで表しているが、これは各塩基を2ビットのデータで表すのと等価である。また、データの割り当ては表1には限定されず、例えばチミンを含むヌクレオチドをbin(00)、アデニンを含むヌクレオチドをbin(11) とするか、又はグアニンを含むヌクレオチドをbin(10)、シトシンを含むヌクレオチドをbin(01) としてもよい。それ以外に、アデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、1対のデータbin(01), bin(10) を割り当て、グアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに1対のデータbin(00), bin(11) を割り当ててもよい。また、RNAの場合には、チミンを含むヌクレオチドに割り当てられているデータをウラシルを含むヌクレオチドに割り当てて、それ以外のヌクレオチドにはDNAのヌクレオチドと同じデータを割り当てればよい。

本例では図2に示すDNA分子5のヌクレオチドの配列情報を扱うものとする。その配列情報は、NCBI (The National Center for Biotechnology Information) より提供されているウェブサイト1 (<ftp://ncbi.nlm.nih.gov/genbank/genomes/bacteria/>) より入手した大腸菌 (*Escherichia coli*: *E. coli*) のDNAの一系列のヌクレオチドの配列の一部である。

図2において、DNA分子5は、1対の重合体の鎖6A、6B (二重らせん) より構成され、一方の重合体の鎖6Aは、アデニンを含むヌクレオチド7A、グアニンを含むヌクレオチド7G、シトシンを含むヌクレオチド7C、及びチミンを含むヌクレオチド7Tよりなる4種類のヌクレオチドの配列であり、他方の重合体の鎖6Bは、鎖6Aに対して相補的なヌクレオチドの配列である。この際に、図1の情報処理装置10には一方の重合体の鎖6Aの配列を示すテキストデータ、即ち”AGCTTT・・・”の文字列のデータが供給される。それに対して、情報処理装置10は、そのテキストデータを後述のようにN行でM列 (N, Mは2以上の整数) のブロックに分割した後、各ブロック中の文

字A, G, C, Tを表1の変換テーブルに基づいて順次2ビットのデータに変換することによって、数値データとしてのバイナリーデータBNA(=bin(0001101111...))を得る。そして、このバイナリーデータBNAが図1の磁気ディスク装置17のワーキングファイル20に記録される。そのバイナリーデータBNAのデータ量は、元のテキストデータの1/4となっている。

この場合、そのワーキングファイル20の先頭の所定数のバイトの領域に、例えばその配列がDNA又はRNAのどちらかを示すデータ(即ち、bin(11))を文字T又は文字Uの何れに解釈するかを示すデータ)、ヌクレオチドの個数を示すデータ、及びその他の必要なデータを記録しておけばよい。また、そのワーキングファイル20の長さが1バイト(8ビット)単位で規定されている場合に、バイナリーデータBNAの末尾で1バイトの端数のデータが生じたときには、予め定めておいたダミーデータを付加すればよい。これでもデータ量は殆ど増加しない。そして、一例としてユーザ(コンピュータシステム2Bの所有者)から供給者(コンピュータシステム2Aの所有者)に対して図2のDNA分子5の配列情報の購入希望が届いたときに、ワーキングファイル20のデータが通信ネットワーク1及び不図示のプロバイダを介して、電子メールの添付ファイルとしてコンピュータシステム2B側に供給される。この際に、そのワーキングファイル20のデータを更に圧縮ファイル(ZIPファイル、又はLHAファイル等)として送信してもよい。この際に、ワーキングファイル20のデータ量はもとのテキストデータのほぼ1/4であるため、元のテキストデータ(更に圧縮ファイルとした場合も同様)自体を送信する場合に比べて送信時間はほぼ1/4となり、供給者側及びユーザ側双方の通信コストが低減できる。

そして、ユーザ側で、その受信したワーキングファイル20のデータから図2の一方の重合体の鎖6Aの配列のテキストデータを復元する場合には、コンピュータシステム2Bにおいて、ワーキングファイル20中のバイナリーデー

タBNAを、表1を用いて文字A, G, C, T (又はU) の何れかに順次逆変換すればよい。また、その際に例えば図2の他方の相補的な重合体の鎖6Bのヌクレオチドの配列を示すテキストデータが必要になった場合には、コンピュータシステム2Bにおいて、図2に示すように、バイナリーデータBNAのビット毎の反転操作を行って反転バイナリーデータNOT(BNA) (=bin(1110010000
5
...))を得る。この反転バイナリーデータNOT(BNA)は、他方の重合体の鎖6Bのヌクレオチドの配列を示すテキストデータ(文字列"TCGAAA...")を表1に従って変換したバイナリーデータBNBと同一である。従って、その反転バイナリーデータNOT(BNA)を、表1を用いて文字A, G, C, T (又はU)
10
の何れかに順次逆変換するのみで、極めて高速に相補的な重合体の鎖6Bの配列のテキストデータを得ることができる。この際に、通常のコンピュータにおいては、ビット毎の反転操作は、極めて高速に実行することができる。なお、そのビット毎の反転操作は、例えばbin(111111...)との排他的論理和演算で代用してもよい。

15 なお、ワーキングファイル20のデータを通信ネットワーク1を介してユーザ側に送信する代わりに、ワーキングファイル20の内容をCD-R/RWドライブ15によってCD-R16に記録し、このCD-R16を郵送等によってユーザ側に供給してもよい。例えば一人の人間の全部のDNAの配列情報(ゲノム)は、テキストデータでは3Gバイト程度になるが、これを表1を用
20 いて本例の数値データとしてのバイナリーデータに変換すると、3/4Gバイト程度、即ち750Mバイト程度になる。現在のCD-R, CD-ROMの容量は約650Mバイトであるため、その750Mバイト程度のバイナリーデータは例えば一部又は全部を圧縮ファイルとすることによって、余裕を持ってCD-R16に記録することができる。これに対して、その750Mバイト程度
25 のデータを通信ネットワーク1を介して送信しようとする、現状でも送信時間がかかり過ぎる場合がある。

また、一つのアミノ酸の種類は一系列の3個のヌクレオチドの配列、即ち一つの遺伝子コドン (codon) によって決定される。そこで、1つのアミノ酸に対応する3個のヌクレオチドをそれぞれ2ビットのデータで表したときに得られる6ビットのデータの中で、最も小さいデータでそのアミノ酸を表すものとする。具体的に、各ヌクレオチドを表1のように表した場合に、いくつかのアミノ酸について得られる6ビットの表現を以下の表2に示す。表1中で<>の中のデータがそのアミノ酸のデータとして使用される。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。

《表2》

10	アミノ酸	遺伝子コドン	6ビットのデータ
	アラニン (Ala)	G C A	<bin (011000)>
		G C G	bin (011001)
		G C C	bin (011010)
		G C U	bin (011011)
15	システイン (Cys)	U G C	<bin (110110)>
		U G U	bin (110111)
	グルタミン酸 (Glu)	G A A	<bin (010000)>
		G A G	bin (010001)
	ヒスチジン (His)	C A C	<bin (100010)>
20		C A U	bin (100011)
	イソロイシン (Ile)	A U A	<bin (001100)>
		A U C	bin (001110)
		A U U	bin (001111)
	リジン (Lys)	A A A	<bin (000000)>
25		A A G	bin (000001)。

次に、本例の情報処理装置10の第2の基本的な処理動作につき説明する。

本例では、ヌクレオチドの配列を示す膨大な量のテキストデータ（又はこれを表 1 に基づいて変換して得られる数値データより、所定のハッシュ関数を用いて数学的な要約値（message digest）を算出する。本例ではそのハッシュ関数として、ライベスト（R. Rivest）によって提案された MD 5 ハッシュ関数を使用する。MD 5 ハッシュ関数のアルゴリズムについては、ネットワークワーキンググループ及びライベストによって開設されているウェブサイト 2 (<http://www.kleinschmidt.com/edi/md5.htm>) に開示されている。或るテキストデータ（テキストファイル）にその MD 5 ハッシュ関数を施すことによって、128 ビットの要約値が得られる。通常のコンピュータでも今後は 64 ビットの CPU が使用されるようになると考えられるが、この場合に 128（＝2・64）ビットの要約値は非常に扱い易い長さである。この場合には、192（＝3・64）ビットの要約値も比較的扱い易いと考えられる。

また、本例では、その MD 5 ハッシュ関数のプログラムとして、そのウェブサイト 2 において公開されている、RSA データセキュリティ社（RSA Data Security Inc.）によって開発されたプログラムを使用した。

その要約値の使用方法の一例として、DNA の配列情報の供給者（情報処理装置 10）は、所定の生物の DNA のヌクレオチドの配列を読み取り、これに対応するテキストデータより、上記のハッシュ関数を用いて要約値を算出し、この要約値をその生物の名称、及び DNA の位置を示す情報と共にインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、その生物の DNA の配列情報を最先に解読したことを主張できると考えられる。その後、或るユーザからのその配列情報の購入希望が来たときに、その供給者は、そのヌクレオチドの配列のテキストデータを表 1 を用いてバイナリーデータに変換し、このバイナリーデータを例えば通信ネットワーク 1 を介して電子メールの形でそのユーザに送信する。これに対してユーザ側では、そのバイナリーデータを表 1 を用いてテキストデータに逆変

換し、この逆変換されたテキストデータに上記のハッシュ関数を施して要約値を求める。

そして、この要約値とその供給者によって公開されている要約値とが等しいときには、購入した配列情報が、供給の保持している配列情報と等しいことが極めて高い確率で保証される。更に、ユーザ側では、複数の供給者が公開している要約値を比較することによって、同じ配列情報を異なる複数の供給者から重複して購入することを防止することができる。これらの際に、ヌクレオチドの配列の長さ、及び先端部や末尾の一部の短い配列の比較を行うことによって、その配列情報の同一性を高めることができる。

また、例えば所定の生物について複数の研究者が並行して研究を行っている場合に、第1の研究者が保持しているヌクレオチドの配列と第2の研究者が保持しているヌクレオチドの配列との同一性を保証する必要がある。この際に、研究対象とするDNAのヌクレオチドの配列数が例えば1億個程度とすると、その配列のテキストデータは100Mバイト程度になる。このような長い2つのテキストデータに対して1文字ずつの比較によって、同一性を確認するのは容易ではない。これに対して本例では、先ず第1の研究者側で、テキストデータの長さ、及びハッシュ関数による要約値を算出し、これを電子メール等で第2研究者側に送信する。これに対して、第2の研究者側でも自分のテキストデータの長さ及びハッシュ関数による要約値を算出し、これらの値を第1の研究者から送信された値と比較することによって、2つの膨大な長さのテキストデータの同一性を容易に高い確率で保証できる。この際にも、更に例えばヌクレオチドの配列の先端部及び末尾の所定長さの配列同士を比較することによって、その同一性を高めることができる。

なお、ハッシュ関数としては、例えば文献2 (FIPS Publication 180, 1993) で開示されているように、NBS (National Bureau of Standards) によって提案されたSHS (Secure Hash Standard) ハッシュ関数を使用してもよい。S

HSハッシュ関数は、MD5ハッシュ関数よりも複雑な演算を行うと共に、160ビットの要約値が得られる。これに関して、例えばタンパク質を構成するアミノ酸の配列数は20個～1000個程度であり、特に一文字表記を使用する際にはそれに対応するテキストデータも20バイト～1kバイト程度に短くなるため、要約値から元のテキストデータが推定し易いと考えられる。そこで、

5 アミノ酸の配列情報の要約値を求める際には、SHSハッシュ関数を使用する方が望ましいことがある。

また、例えばヌクレオチドの配列を示す2つの膨大な長さのテキストデータの同一性を確認するために、ハッシュ関数の要約値を算出するような場合には、

10 それ程複雑な計算を繰り返して行う必要は無いと考えられる。そこで、このような用途では、例えば文献3 (R. L. Rivest: "The MD4 message digest algorithm", Lecture Notes in Computer Science, 537, 303-311 (1991)) で開示されているMD4ハッシュ関数を使用してもよいと考えられる。また、そのように単に同一性を確認する用途では、要約値の長さも40ビット～128ビット程度でよい場合がある。

15

次に、本例のDNA情報の供給者（コンピュータシステム2A）と、ユーザ（コンピュータシステム2B）との間でDNAの配列情報を受け渡す際のビジネスモデルの一例につき図3～図6のフローチャートを参照して詳細に説明する。まず、DNA情報の供給者側では、図3のステップ101において、シー

20 ケンサー4を使用して標準となる試料（標準試料Eとする）のDNA中の一方の系列のヌクレオチドの配列を読み取り、読み取った配列を表すテキストデータTX1を情報処理装置10に供給する。本例では、その標準試料Eを大腸菌として、そのテキストデータTX1として、図7に示すように、上記のウェブサイト1から入手した大腸菌のDNAの配列情報の内の、最初から2048個

25 までのヌクレオチドの配列を示すテキストデータを使用する。

標準試料EのDNA配列は配列番号1に示されている。図7のテキストデー

タは、配列番号 1 の配列から数字データを除いて、a, g, c, t の文字をそれぞれ A, G, C, T で置き換えたものに相当する。

次のステップ 102 において、情報処理装置 10 は、供給されたテキストデータ TX1 に上記の MD5 ハッシュ関数を施して 128 ビットの要約値 AB1 を求めると共に、そのヌクレオチドの配列の数 NA1、及び先頭と末尾との 8 個ずつのヌクレオチドの配列 ST1, SB1 を求める。テキストデータ TX1 に対する具体的な値は下記の通りである。

AB1 = hex (849339ac244cde42b5346ab5989aab61) ... (11)

NA1 = 2048

10 ST1 = AGCTTTTC, SB1 = CGCGAAGG

次のステップ 103 において、情報処理装置 10 は、テキストデータ TX1 を逆方向に並べ替えたテキストデータ TXR1 (=GGAAGC...TCGA) を求め、このテキストデータ TXR1 の MD5 ハッシュ関数による要約値 ABR1、及びこのテキストデータ TXR1 の先頭と末尾との 8 個ずつのヌクレオチドの配列 STR1, SBR1 を求める。配列 STR1, SBR1 は、上記の配列 SB1, ST1 をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。

15 ABR1 = hex (4eb1feae30f522642b912ce3ea09652b) ... (12)

STR1 = GGAAGCGC, SBR1 = CTTTTTCGA

20 次のステップ 104 において、情報処理装置 10 は、標準試料 E の名前の情報（試料を特定する情報）、配列の数 NA1、テキストデータ TX1、配列 ST1, SB1、要約値 AB1、逆方向の配列 STR1, SBR1、及び逆方向の要約値 ABR1 を磁気ディスク装置 17 のマスターファイル 19 に記録する。この際に、マスターファイル 19 を複数のファイルとして、テキストデータ TX1 と、それ以外のデータとを別のファイルに記録してもよい。また、テキストデータ TX1 が例えば 100 M バイト程度以上になる場合には、テキストデ

25

ータTX1を複数のマスターファイルに分割して記録してもよい。

次のステップ105において、情報処理装置10は、図7に示すように、標準試料EのテキストデータTX1を配列方向（ヌクレオチドの配列方向）にN行で、その配列方向に直交する方向（以下、「非配列方向」という）にM列の
5 16文字の長さの部分テキストデータT(i, j) (i = 1 ~ N, j = 1 ~ M)に分割する。なお、N, Mはそれぞれ2以上の任意の整数であり、(1)式、
(2)式を用いて既に説明したように、テキストデータTX1が100kバイト程度（又はこの整数倍）であるときに、このテキストデータTX1に対して
1 / 100程度のデータ量のシンδροーム情報を得たい場合には、例えばNの
10 値が64、Mの値が128に設定される。以下では説明を簡単にするために、
テキストデータTX1を4行で、かつ32列に分割した場合を想定する。即ち、
N = 4, M = 32とする。この場合、本例では端数は生じないが、例えば図7
において、最後の部分テキストデータT(4, 32)中の文字が16個より少ない場合には、足りない部分には予め定めた文字（例えば文字A）をダミーデ
15 ータとして付加すればよい。また、部分テキストデータT(i, j)の長さは、
16文字以外の任意の長さでよいが、処理速度を高めるためには、8文字の倍数が効率的である。

更に、情報処理装置10は、図7の各部分テキストデータT(i, j)を表
1（変換テーブル）に基づいてそれぞれ32ビットのバイナリーデータ（数値
20 データ）よりなる変換データA(i, j)に変換する。この結果、図8に示す
4行で、32列の変換データA(i, j)（16進数表示）が得られる。また、
変換データA(i, j)を対応するヌクレオチドの配列方向に連続して配列したときの集合体（数値データ）をバイナリーデータBN1とする。このバイナ
リーデータBN1は、図2の一方のバイナリーデータBNAと同じものである
25 が、図2のバイナリーデータBNAは2進数表示されており、図8のバイナ
リーデータ（変換データA(i, j)）は16進数表示されている。この場合、

各部分テキストデータ $T(i, j)$ の長さは 16 バイト (= 128 ビット) であるため、図 7 の全体のテキストデータ $TX1$ に対して、図 8 の全体のバイナリーデータ $BN1$ のデータ量は $1/4$ に減少している。なお、図 7 の部分テキストデータ $T(i, j)$ と図 8 の変換データ $A(i, j)$ とは等価であるため、
 5 上記の方法の代わりに、元のテキストデータ $TX1$ を表 1 に基づいてバイナリーデータ $BN1$ に変換した後、このバイナリーデータ $BN1$ を N 行で、 M 列の変換データ $A(i, j)$ に分割してもよい。

次のステップ 106 において、情報処理装置 10 は、図 8 の全部の変換データ $A(i, j)$ の内で、各列の変換データ $A(i, j)$ の配列方向に対する法 $2^{32} \pmod{2^{32}}$ のもとでの和、即ち配列方向のシンδροーム (syndrome) $C(j)$ ($j = 1 \sim 32$) を計算する。 $C(j)$ は以下のように表すことができる。

$$C(j) = A(1, j) + A(2, j) + \dots + A(4, j) \pmod{2^{32}} \quad \dots (13)$$

更に、情報処理装置 10 は、各行の変換データ $A(i, j)$ ($i = 1 \sim 4$) の内で奇数番目の変換データ $A(i, 2j' - 1)$ ($j' = 1 \sim 16$) の非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データ $A(i, 2j')$ の非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向のシンδροーム $B1(i)$, $B2(i)$ ($i = 1 \sim 4$) を次式より計算する。

$$B1(i) = A(i, 1) + A(i, 3) + \dots + A(i, 31) \pmod{2^{32}} \quad \dots (14)$$

$$B2(i) = A(i, 2) + A(i, 4) + \dots + A(i, 32) \pmod{2^{32}} \quad \dots (15)$$

変換データ $A(i, j)$ に対する実際の計算結果が、図 8 のシンδροーム $C(j)$, $B1(i)$, $B2(i)$ として表示されている。

また、図 9 は、図 8 の標準試料 E のデータ中からシンδροーム $C(j)$, $B1(i)$, $B2(i)$ だけを取り出して表示したものである。この例においては、シンδροーム $C(j)$, $B1(i)$, $B2(i)$ はそれぞれ 32 ビット (4 バイト) であるため、全部のシンδροームのデータ量は、160 (= 4 ·

40) バイトとなる。従って、全部のシンδροームのデータ量は、図7の全体のテキストデータTX1 (2048バイト) に対してほぼ1/13に減少しており、図8の全体のバイナリーデータBN1に対してもほぼ1/3に減少している。

5 次に図4のステップ107において、情報処理装置10は、標準試料Eの名前の情報、配列の数NA1、バイナリーデータBN1、シンδροームC(j), B1(i), B2(i)を磁気ディスク装置17のワーキングファイル20に記録する。この際に、ワーキングファイル20を複数のファイルとして、バイナリーデータBN1と、シンδροームC(j), B1(i), B2(i)とを別のファイルに記録してもよい。更に、バイナリーデータBN1と共に、ステップ102で算出した要約値AB1をワーキングファイル20に記録してもよい。

また、バイナリーデータBN1が長いときには、バイナリーデータBN1を複数のファイルに分割して記録してもよい。更に、図7のテキストデータTX1 (ひいては図8のバイナリーデータBN1) がかなり長い場合には、テキストデータTX1を例えば100kバイト程度を単位として複数のデータ群に分割し、各データ群毎にシンδροームC(j), B1(i), B2(i)を求めるようにしてもよい。

更に、ステップ107において、DNA情報の供給者は、ワーキングファイル20に記録した情報、即ち標準試料Eの名前の情報、配列の数NA1、バイナリーデータBN1、シンδροームC(j), B1(i), B2(i)と、マスターファイル17に記録した要約値AB1, ABR1の情報とを、CD-R/RWドライブ15を介してCD-R16に記録してもよい。このCD-R16から、更に多数のCD-ROMを作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

次の、ステップ108において、情報処理装置10は、標準試料Eの名前の

情報、配列の数NA 1、配列ST 1、SB 1、要約値AB 1、逆方向の配列STR 1、SBR 1、及び逆方向の要約値ABR 1を磁気ディスク装置17のコンテンツファイル21に記録する。図7のテキストデータTX 1が仮に100Mバイト程度の膨大なものであっても、コンテンツファイル21に記録されるデータは500バイト程度の僅かなものである。更に、情報処理装置10は、コンテンツファイル21中の情報を通信ネットワーク1を介してコンテンツのプロバイダ3に送信する。これによって、コンテンツファイル21中の情報はプロバイダ3のサーバ内の閲覧可能なコンテンツファイル31に記録されて、第3者がインターネットを介して自由に閲覧できるようになる。

次のステップ109において、DNA情報の供給者は、ユーザから購入要求が来るのを待つ状態となる。そして、(a) ユーザから標準試料Eに対する簡易データの要求があったときには、ステップ110に移行して、情報処理装置10は、磁気ディスク装置17のワーキングファイル20の中のシンドロームC(j)、B1(i)、B2(i)の情報を例えば電子メールの添付ファイルとしてそのユーザに送信する。一方、ステップ109において、(b) ユーザから完全データの要求があったときには、ステップ111に移行して、情報処理装置10は、ワーキングファイル20中のバイナリーデータBN1をZIPファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値AB1を同時に送信してもよい。本例によれば、簡易データ(シンドローム)はデータ量が少ないために短時間で送信することができる。また、完全データ(バイナリーデータBN1)でも元のテキストデータに比べて1/4のデータ量であるため、比較的短時間に送信することができる。

また、ステップ109において、ユーザは、必要に応じて部分データ、即ち図8の全部の変換データA(i, j)の内の所望のデータ、例えば2つの変換データA(4, 16)及びA(1, 17)のみをその供給者から購入するよう

にしてもよい。これによって、必要な正確なデータのみを短時間に入手することができる。

次に、DNA情報のユーザ（図1のコンピュータシステム2Bの所有者とする）側では、図5のステップ121において、図1の通信ネットワーク1（インターネット）を介してプロバイダ3のサーバ内のコンテンツファイル31の内容を閲覧し、その中からステップ108で供給者から送信された情報、即ち標準試料Eの名前の情報、ヌクレオチドの配列の数NA1、配列ST1、SB1、要約値AB1、逆方向の配列STR1、SBR1、及び逆方向の要約値ABR1を読み取り、読み取った情報をコンピュータシステム2B内の記憶装置の一時ファイルに記録する。

次の、ステップ122において、そのユーザは、不図示のDNAのシーケンサーを用いて、標準試料Eと同じ種類で検査対象の試料FのDNA中の一方の系列のヌクレオチドの配列を読み取り、読み取られた配列を示すテキストデータTX2をコンピュータシステム2B内の情報処理装置に取り込む。その検査対象の試料Fとは、例えば突然変異を起こしていると思われる大腸菌であり、そのテキストデータTX2は、標準試料EのテキストデータTX1と同様に最初から2048個までのヌクレオチドの配列を示すものとする。

試料FのDNA配列は配列番号2に示されている。後述の図10のテキストデータは、配列番号2の配列から数字データを除いて、a, g, c, tの文字をそれぞれA, G, C, Tで置き換えたものに相当する。

図10は、その試料FのDNAのヌクレオチドの配列に対応するテキストデータTX2を示し、この図10の配列の内のアンダーラインを付した部分のみが、図7の標準試料Eの配列と異なっている。即ち、試料Fの配列は、標準試料Eの部分テキストデータT(4, 16), T(1, 17)の部分だけが以下のように異なっている。なお、この段階では、ユーザは、試料Fの配列と標準試料Eの配列とのどの部分が相違しているのかは分からない。

標準試料 E

試料 F

T (4, 16) = ATTTGGACGGACGTTG → ATTTGGACATTATGGC

T (1, 17) = ACGGGGTCTATACCTG → GGCCAACTTATACCTG

そして、ユーザのコンピュータシステム 2 B 側の情報処理装置においても、
 5 DNA の配列情報を処理するためのアプリケーション・プログラムが起動され
 ている。そして、その情報処理装置は、ステップ 1 2 3 において、読み取られ
 たテキストデータ TX 2 に上記の MD 5 ハッシュ関数を施して 1 2 8 ビットの
 要約値 AB 2 を求めると共に、そのヌクレオチドの配列の数 NA 2、及び先頭
 と末尾との 8 個ずつのヌクレオチドの配列 ST 2、SB 2 を求め、これらを内
 10 部の記憶装置の第 1 データファイルに記録する。テキストデータ TX 2 (図 1
 0) に対する具体的な値は下記の通りである。

AB 2 = hex (1457b51222a83c3222e87cb4d4e63305) … (1 6)

NA 2 = 2 0 4 8

ST 2 = A G C T T T T C, SB 2 = C G C G A A G G

15 次のステップ 1 2 4 において、情報処理装置は、試料 F の配列数 NA 2 と標
 準試料 E の配列数 NA 1 とが等しいかどうかを調べ、両者が異なっている場合
 には、ユーザはステップ 1 2 5 に移行して、別の DNA 情報を検索し、NA 2
 と同じ配列数の DNA 情報をサーチする。本例では、ステップ 1 2 4 において、
 NA 2 = NA 1 であるため、動作はステップ 1 2 6 に移行して、試料 F の先頭
 20 と末尾との一部の配列 ST 2、SB 2 が、標準試料 E の配列 ST 1、SB 1 と
 等しいかどうか、更に試料 F の要約値 AB 2 が標準試料 E の要約値 AB 1 (ス
 テップ 1 2 1 で一時ファイルに記録されている) と等しいかどうかを調べる。
 これらが共に等しい場合には、試料 F の配列と標準試料 E の配列とは非常に高
 い確率 (ほぼ $1 / 2^{128} \cong 1 / 10^{38}$ 程度の確率) で一致しているとみなすこ
 25 とができる。従って、ステップ 1 2 7 に移行して、コンピュータシステム 2 B
 の情報処理装置は、その第 1 データファイルに「試料 F の DNA 構造は、標準

試料EのDNA構造と同一」との情報を記録する。

但し、本例では、 $ST2 = ST1$ 、 $SB2 = SB1$ が成立するが、(11)式及び(16)式より $AB2 \neq AB1$ であるため、動作はステップ126からステップ128に移行して、その情報処理装置は、試料Fの先頭と末尾との一部の配列 $ST2$ 、 $SB2$ が、標準試料Eを逆に並べた配列の一部の配列 $STR1$ 、 $SBR1$ と等しいかどうか、更に試料Fの要約値 $AB2$ が標準試料Eを逆に並べた配列の要約値 $ABR1$ と等しいかどうかを調べる。これらが共に等しい場合には、試料Fの配列と標準試料Eを逆に並べた配列とは非常に高い確率で一致しているとみなすことができる。従って、ステップ139に移行して、コンピュータシステム2Bの情報処理装置は、その第1データファイルに「試料FのDNA構造は、標準試料EのDNA構造と回文 (palindrome) の関係にある」との情報を記録する。

本例では、 $ST2 \neq STR2$ 、 $SB2 \neq SBR2$ 、かつ(12)式及び(16)式より $AB2 \neq ABR1$ であるため、動作はステップ128からステップ129に移行して、そのユーザは、通信ネットワーク1 (インターネット) を介してDNA情報の供給者から上記の簡易データ、即ち標準試料Eのシンドローム $C(j)$ 、 $B1(i)$ 、 $B2(i)$ の情報(図9の情報)を購入し、購入した情報をコンピュータシステム2B (情報処理装置) 内の記憶装置の第2データファイルに記録する。

次に、図6のステップ130において、コンピュータシステム2Bの情報処理装置は、図10に示すように、試料Fのテキストデータ $TX2$ を配列方向 (ヌクレオチドの配列方向) にN行で、非配列方向にM列の16文字の長さの部分テキストデータ $TF(i, j)$ ($i = 1 \sim N$, $j = 1 \sim M$) に分割する。分割数N, Mは標準試料Eの分割数と同じであり、本例では、 $N = 4$, $M = 3$ 2である。更に、情報処理装置は、図10の各部分テキストデータ $TF(i, j)$ を表1 (変換テーブル) に基づいてそれぞれ32ビットのバイナリーデー

タ（数値データ）よりなる変換データ $AF(i, j)$ に変換する。この結果、図 11 に示す 4 行で、32 列の変換データ $AF(i, j)$ （16 進数表示）が得られる。また、変換データ $AF(i, j)$ を連続して配列した集合体（数値データ）をバイナリーデータ $BN2$ とする。

- 5 次に、情報処理装置は、ステップ 106 の動作と同様にして、図 11 の全部の変換データ $AF(i, j)$ の内で、各列の変換データ $AF(i, j)$ の配列方向に対する法 $2^{32} \pmod{2^{32}}$ のもとでの和、即ち配列方向のシンδροーム $CF(j)$ （ $j = 1 \sim 32$ ）を計算する。 $CF(j)$ は、(13) 式で $A(i, j)$ を $AF(i, j)$ で置き換えた式で計算される。更に、情報処理装置は、各行の変換データ $AF(i, j)$ （ $i = 1 \sim 4$ ）の内で奇数番目の変換データ $AF(i, 2j' - 1)$ （ $j' = 1 \sim 16$ ）の非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データ $AF(i, 2j')$ の非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向のシンδροーム $B1F(i)$ 、 $B2F(i)$ （ $i = 1 \sim 4$ ）を計算する。 $B1F(i)$ 、 $B2F(i)$ は、
- 10 (14) 式、(15) 式で $A(i, j)$ を $AF(i, j)$ で置き換えた式で計算される。変換データ $AF(i, j)$ に対する実際の計算結果が、図 11 のシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ として表示されている。
- 15 (14) 式、(15) 式で $A(i, j)$ を $AF(i, j)$ で置き換えた式で計算される。変換データ $AF(i, j)$ に対する実際の計算結果が、図 11 のシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ として表示されている。

- 20 図 8（標準試料 E）と図 11（試料 F）とを比較すると、図 8 の変換データ $A(4, 16)$ 、 $A(1, 17)$ に対して図 11 の変換データ $AF(4, 16)$ 、 $AF(1, 17)$ の値が異なっている。従って、それに対応して図 11 にアンダーラインを付して示すように、図 11 の配列方向の 2 つのシンδροーム $CF(16)$ 、 $CF(17)$ 、及び非配列方向の 2 つのシンδροーム $B1F(1)$ 、 $B2F(4)$ の値が、図 8 の対応するシンδροーム $C(16)$ 、 $C(17)$ 、 $B1(1)$ 、 $B2(4)$ の値と異なっている。

- 25 また、図 12 は、主に図 11 の試料 F のデータ中からシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ だけを取り出して表示したものである。

次に、ステップ 1 3 1 において、その情報処理装置は、供給者から購入した簡易データの 1 組のシンδροーム、即ち図 8（標準試料 E）の 1 組のシンδροーム C（j），B 1（i），B 2（i）と、上記のように求めた試料 F の 1 組のシンδροーム C F（j），B 1 F（i），B 2 F（i）とを比較して、相違するシンδροームをサーチする。本例では、図 1 1 のシンδροーム C F（1 6），C F（1 7）、及びシンδροーム B 1 F（1），B 2 F（4）が相違するシンδροームとして特定される。この場合、配列方向の相違するシンδροーム C F（1 6），C F（1 7）の列と、非配列方向の相違するシンδροーム B 1 F（1），B 2 F（4）の行との交点が、標準試料 E に対して相違する変換データの位置となる。従って、図 1 1 の第 4 行で第 1 6 列の変換データ A F（4，1 6）、及び第 1 行で第 1 7 列の変換データ A F（1，1 7）が相違する変換データとして特定される。

次のステップ 1 3 2 において、その情報処理装置は、図 1 1 の変換データ A F（i，j）中で図 8 の変換データ A（i，j）と相違する変換データ（A F（i'，j'）とする）は、各行、又は各列に多くとも一つかどうかを調べる。これが成立する場合には、その変換データ A F（i'，j'）に対応する標準試料 E の変換データは、法 2^{32} のもとでの連立方程式によって容易に求めることができる。本例では、それが成立する、即ち相違する変換データは、第 1 行、第 4 行に一つずつで、かつ第 1 6 列、第 1 7 列に一つずつであるため、動作はステップ 1 3 3 に移行する。そして、その情報処理装置は、先ず変換データ A F（4，1 6）から標準試料 E の変換データ A（4，1 6）を復元するために、図 8 のシンδροーム C（1 6）、図 1 1 のシンδροーム C F（1 6）、及び変換データ A F（4，1 6）を用いて次の演算を行う。

$$\begin{aligned}
 A(4, 16) &= C(16) - CF(16) + AF(4, 16) \pmod{2^{32}} \\
 &= \text{hex}(7c33894d) - \text{hex}(7c3373a6) + \text{hex}(3f523cd6) \pmod{2^{32}} \\
 &= \text{hex}(3f52527d) \quad \dots (17)
 \end{aligned}$$

この結果を図8の変換データA(4, 16)と比較すると、復元が正確に行われていることが分かる。

続いて、情報処理装置は、変換データAF(1, 17)から標準試料Eの変換データA(1, 17)を復元するために、図8のシンδροームC(17)、
 5 図11のシンδροームCF(17)、及び変換データAF(1, 17)を用いて次の演算を行う。

$$\begin{aligned} A(1, 17) &= C(17) - CF(17) + AF(1, 17) \pmod{2^{32}} \\ &= \text{hex}(31b4c2ad) - \text{hex}(6661c2ad) + \text{hex}(5a0bccad) \pmod{2^{32}} \\ &= \text{hex}(255eccad) \quad \dots (18) \end{aligned}$$

10 この結果を図8の変換データA(1, 17)と比較すると、復元が正確に行われていることが分かる。また、復元された変換データA(4, 16), A(1, 17)が、図12中の試料FのシンδροームCF(j), B1F(i), B2F(i)の内側に表示されている。図12の変換データA(4, 16), A(1, 17)を表1に従って逆変換して得られる部分テキストデータは、図
 15 7の標準試料Eの部分テキストデータT(4, 16), T(1, 17)と等しいことが分かる。

次のステップ134において、その情報処理装置は、復元された変換データA(i', j'), 即ちA(4, 16), A(1, 17)で、図11の試料FのバイナリーデータBN2中の対応する変換データAF(4, 16), AF
 20 (1, 17)を置き換えた後、この置き換えによって得られるバイナリーデータBN2を表1に基づいてテキストデータTX1'に逆変換する。更に情報処理装置は、そのテキストデータTX1'よりMD5ハッシュ関数を用いて128ビットの要約値AB1'を算出し、この要約値AB1'が標準試料Eの要約値AB1(ステップ121で一時ファイルに記録されている)と等しいかどうかを確認する。本例では、AB1' = AB1が成立するが、例えば図11の試
 25 料Fの変換データAF(i, j)中の相違する変換データの位置や内容によっ

ては、その相違する位置がステップ 1 3 2 で正確に検出されない可能性がある。
このような場合に、 $AB1' \neq AB1$ となったときには、ステップ 1 3 5 に移行すればよい。通常は、 $AB1' = AB1$ が成立して、動作はステップ 1 3 8 に移行して、情報処理装置は、上記の第 1 データファイルに「試料 F の配列と標準試料 E の配列との内で相違する部分の位置 (i' , j')、及び相違する部分テキストデータの対」の情報を記録する。本例では、位置 (i' , j') として位置 (4, 16), (1, 17) が、相違する部分テキストデータの対として A (4, 16), AF (4, 16) 及び A (1, 17), AF (1, 17) が記録される。

一方、ステップ 1 3 2 において、相違する変換データ AF (i' , j') が少なくとも一行に 2 個以上で、かつ列方向にも 2 個以上 (奇数番目又は偶数番目で 2 個以上を意味する) 存在する場合には、変換データの正確な復元は困難である。そこで、動作はステップ 1 3 5 に移行して、そのユーザはその DNA 情報の供給者から標準試料 E の完全データ、即ち図 8 のバイナリーデータ BN 1 を通信ネットワーク 1 (インターネット) を介して購入し、コンピュータシステム 2 B の情報処理装置は、そのバイナリーデータ BN 1 を記憶装置の第 3 データファイルに記録する。

次のステップ 1 3 6 において、その情報処理装置は、そのバイナリーデータ BN 1 を表 1 に基づいてテキストデータ TX 1' に逆変換 (復元) し、そのテキストデータ TX 1' より MD 5 ハッシュ関数を用いて 128 ビットの要約値 AB 1' を算出し、この要約値 AB 1' が標準試料 E の要約値 AB 1 (ステップ 1 2 1 で一時ファイルに記録されている) と等しいかどうかを確認する。通常は、 $AB1' = AB1$ が成立するが、例えば通信エラー等によって送信されたバイナリーデータ BN 1 の中にエラーが生じている場合には、 $AB1' \neq AB1$ となる。このときには、例えば供給者に完全データの再送信を行う等の対処を行う。そして、ステップ 1 3 6 で $AB1' = AB1$ が成立するときには、

ステップ137に移行して情報処理装置は、標準試料EのバイナリーデータBN1中で、試料Fの相違している変換データAF(i', j')に対応する変換データA(i', j')を求める。その後、動作はステップ138に移行する。

- 5 このように本例のビジネスモデルによれば、第1段階として標準試料Eの要約値AB1と試料Fの要約値AB2とを比較して、両者が等しいときには試料FのDNAの構造は標準試料EのDNAの構造と同じとみなすため、DNA情報の供給者からそれ以上の情報を購入する必要が無い。また、第2段階として、標準試料EのシンδροームC(j), B1(i), B2(i)と試料Fのシン
- 10 ドロームCF(j), B1F(i), B2F(i)とを比較して、相違する変換データAF(i, j)の個数が少ない場合には、対応する標準試料Eの変換データA(i, j)を復元するため、膨大な完全データを購入する必要がなく、情報処理コストを低減できる。

- 15 なお、上記のステップ135では、ユーザはDNA情報の供給者から完全データ（バイナリーデータBN1）を購入しているが、別の方法として、ステップ131で特定された相違する変換データAF(i', j')に対応する標準試料Eの変換データA(i', j')のみを購入してもよい。これによって、通信コストを低減できる。

- 20 また、本例のシンδροームの使用方法に関して、本例では非配列方向（列方向）に2組のシンδροームB1(i), B2(i)を求めているため、図11の試料Fの変換データAF(i, j)において、連続する2列の変換データAF(i, j)、例えばAF(i, 16), AF(i, 17) (i=1~4)の全部（8個）が標準試料Eの変換データA(i, j)と相違していても、その相違する部分（エラーコード）の位置を正確に検出することができる。更に、
- 25 非配列方向のシンδροームB1F(i), B2F(i)、及び変換データAF(i, 16), AF(i, 17) (i=1~4)を用いて連立方程式を解くこ

とによって、対応する標準試料Eの変換データA (i, j) の全部を正確に復元できる。即ち、ヌクレオチドの配列方向に対して隣接する2列に跨るような比較的長いエラーコード（バーストエラー）が生じても、本例のシンδροームによってその位置の検出、及び対応する配列の復元を行うことができる。

5 また、本例のシンδροームを用いれば、SNP（一塩基変位多型：Single Nucleotide Polymorphism）のように所定の範囲内で1つのヌクレオチド（塩基）だけが異なっているようなエラーコードは、更に容易にその位置の検出、及び復元を行うことができる。そして、所定の範囲内で、即ち図11の配列中で1つ（ヌクレオチドの1つ分）だけ生じたエラーコードの検出、及び復元を行えば
10 良い場合には、非配列方向のシンδροームB1F (i), B2F (i) の代わりに、それらの和（BF (i) とする）を使用するのみで十分である。この場合には、図8の標準試料Eについても、非配列方向のシンδροームB1 (i), B2 (i) の代わりに、それらの和（B (i) とする）を用意するのみでよい。

15 また、例えば図8（図11でも同様）において、配列方向のシンδροームC (j) を、各列で前半の1組の変換データと後半の1組の変換データとで2つ設け、非配列方向のシンδροームB1 (i), B2 (i) を一つのB (i) とした場合にも、上記の隣接する2列に跨るバーストエラーの検出及び復元を行うことができる。また、より多くのエラーコードの復元を行うためには、計算
20 は極めて複雑になるが、シンδροームC (j), B1 (i), B2 (i) の代わりに、シンδροーム情報として例えばリードソロモンのCRC符号（Reed-Solomon Cyclic Redundancy Check Code: RSCRC Code）を使用してもよい。RSCRC符号については、例えば文献（James S. Plank: Software-Practice & Experience, 27 (9), September, pp. 995-1012 (1997)）で開示されている。

25 なお、上記の実施の形態では、DNA又はRNAを構成するヌクレオチドは4種類であるため、テキストデータTX1をバイナリデータBN1に変換す

る際に、表 1 に示すように各ヌクレオチドを 2 ビットのデータで表している。
これに対して、ヌクレオチド（又は塩基）を表すテキストデータとして、以下
のような 16 種類の文字 a ~ n（8 ビットのアスキーデータ）が使用されるこ
とがある。

- | | | |
|----|---|---|
| 5 | a | アデニン（アデニンを含むヌクレオチドと同義、以下同様） |
| | c | シトシン |
| | g | グアニン |
| | t | チミン |
| | u | ウラシル |
| 10 | m | アデニン、又はシトシン |
| | r | グアニン、又はアデニン |
| | w | アデニン、又はチミン（若しくはウラシル） |
| | s | グアニン、又はシトシン |
| | y | チミン（若しくはウラシル）、又はシトシン |
| 15 | k | グアニン、又はチミン（若しくはウラシル） |
| | v | アデニン、グアニン、又はシトシン |
| | h | アデニン、シトシン、又はチミン（若しくはウラシル） |
| | d | アデニン、グアニン、又はチミン（若しくはウラシル） |
| | b | グアニン、シトシン、又はチミン（若しくはウラシル） |
| 20 | n | （アデニン、シトシン、グアニン、又はチミン（若しくはウラシル））
又は（不明若しくは他の塩基）。 |

この場合には、これら 16 種類の文字を互いに異なる 4 ビットのコードに変
換し、このコードを用いてテキストデータを数値データ（バイナリーデータ）
に変換してもよい。これによって、データ量を 1 / 2 にすることができる。ま
た、将来的にヌクレオチド（塩基）の種類が増加したような場合には、これら
のヌクレオチドを 5 ビット、又は 6 ビットのデータで表現するようにしてもよ

い。

また、上記の実施の形態では、図 7 及び図 10 のヌクレオチドの配列を示すテキストデータよりハッシュ関数によって要約値を算出しているが、情報量としては、それらのテキストデータは図 8 及び図 11 のバイナリーデータ（数値データ）と等価である。従って、これらのバイナリーデータよりハッシュ関数によってそれぞれ要約値を算出し、これらの算出結果同士を比較するようにしてもよい。バイナリーデータはテキストデータに対して 1/4 程度であるため、要約値を算出する時間が短縮できる利点がある。

なお、上記の実施の形態では、DNA 又は RNA 中のヌクレオチドの配列（又は塩基の配列）の情報を処理対象としているが、本発明は、遺伝子を形成するヌクレオチドの配列の情報を処理する場合にも適用できることは言うまでもない。

次に、本発明の実施の形態の他の例につき説明する。本例は、タンパク質又はペプチドを構成するアミノ酸の配列情報を処理する場合に本発明を適用したものである。

本例でも基本的に図 1 のコンピュータシステム 2A を使用できるが、DNA のシーケンサー 4 の代わりに、タンパク質のアミノ酸の配列を決定する配列読み取り装置としてのタンパク質用のシーケンサー（protein Sequencer）が情報処理装置 10 に接続される点が異なっている。なお、その配列読み取り装置としては、アミノ酸の配列のデータベースも使用できる。本例でも、例えば新規の試料 G のタンパク質のアミノ酸の配列をそのシーケンサーで解読した場合に、その配列を示すテキストデータ（TX3 とする）が情報処理装置 10 に供給される。本例では一文字表記を採用するものとして、n 個のアミノ酸の配列に対応するテキストデータは、n バイトの長さである。本例では、その試料 G を大腸菌として、そのテキストデータ TX3 として、図 13 に示すように、上記のウェブサイト 1 から入手した大腸菌の或るタンパク質の 820 個のアミノ酸の

配列を示すテキストデータを使用する。

試料Gのアミノ酸配列は配列番号3に示されている。図13のテキストデータは、配列番号3の配列から数字データを除いて、その配列を一文字表記で表したものに相当する。また、図13においては、そのテキストデータが配列方向（アミノ酸の配列方向）に8行で、その配列方向に直交する非配列方向に26列の4文字の長さの部分テキストデータに分割されており、861番以上のアミノ酸を示すデータ（テキストデータTX3には正確には含まれない部分）の位置には仮に0が表示されている。

次に、情報処理装置10は、供給されたテキストデータTX3に上記のMD5ハッシュ関数を施して128ビットの要約値AB3を求めると共に、そのアミノ酸の配列の数NA3、及び先頭と末尾との8個ずつのアミノ酸の配列ST3、SB3を求める。テキストデータTX3に対する具体的な値は下記の通りである。

AB3=hex(0f66dc2b3024a9739d0e912fde12b8ba) ... (19)

NA3=820

ST3=MRVLKFGG, SB3=TL SWKLGV

次に、情報処理装置10は、テキストデータTX3を逆方向に並べ替えたテキストデータTXR3(=VGLKWS・・・FKLVRM)を求め、このテキストデータTXR3のMD5ハッシュ関数による要約値ABR3、及びこのテキストデータTXR3の先頭と末尾との8個ずつのアミノ酸の配列STR3、SBR3を求める。配列STR3、SBR3は、上記の配列SB3、ST3をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。テキストデータTXR3の配列は元のテキストデータTX3に対して回文(palindrome)の関係にあるとすることができる。

ABR3=hex(e895f433e1e77f84b3cadeead1a52380) ... (20)

STR3 = VGLKWSLT, SBR3 = GGFKLVRM

次に、情報処理装置10は、試料Gの名前の情報（試料を特定する情報）、配列の数NA3、テキストデータTX3、配列ST3、SB3、要約値AB3、逆方向の配列STR3、SBR3、及び逆方向の要約値ABR3を磁気ディスク装置17のマスターファイル19に記録する。この際に、マスターファイル19を複数のファイルとして、テキストデータTX3と、それ以外のデータとを別のファイルに記録してもよい。続いて、情報処理装置10は、例えば図7と同様に図13に示すように、試料GのテキストデータTX3を配列方向（アミノ酸の配列方向）にN行で、その配列方向に直交する非配列方向にM列の4文字の長さの部分テキストデータに分割する。なお、N、Mはそれぞれ2以上の任意の整数であり、一例としてN=16、M=13などとしてもよい。本例ではテキストデータTX3に例えば12文字分のダミーデータ（本例では0を用いるが、例えば文字Aなども使用できる）を付加して得られる832（=4・16・13）バイトのテキストデータ（これをTX3'と呼ぶ）を用い、テキストデータTX3'を一例としてN=8、M=26で分割する。本例では、ヌクレオチドの配列を扱う場合と異なり、その4文字分の部分テキストデータをそのまま32ビットの変換データとして扱う。なお、この際に表2に示すように、各アミノ酸を6ビットのデータで表してもよいが、データ量は3/4程度になるだけであるため、本例では部分テキストデータをそのまま変換データ（数値データ）として扱う。

それに続いて、情報処理装置10は、その8行で26列の変換データに対して、図8の例と同様に、各列の変換データの配列方向に対する法 2^{32} （mod 2^{32} ）のもとでの和、即ち配列方向のシンδροームを計算する。更に、各行の変換データの中で奇数番目の変換データの非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データの非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向の2組のシンδροームを計算する。この例においては、シンド

ロームそれぞれ32ビット（4バイト）であるため、全部のシンドロームのデータ量は、168（ $=4 \cdot 42$ ）バイトとなる。従って、全部のシンドロームのデータ量は、全体の元のテキストデータTX3（820バイト）に対してほぼ1/4～1/5に減少している。

5 次に、情報処理装置10は、試料Gの名前の情報、配列の数NA3、テキストデータTX3、要約値AB3、ABR3、及びシンドロームを磁気ディスク装置17のワーキングファイル20に記録する。この際に、ワーキングファイル20を複数のファイルとしてもよい。その後、情報処理装置10は、試料Gの名前の情報、配列の数NA3、配列ST3、SB3、要約値AB3、逆方向
10 の配列STR3、SBR3、及び逆方向の要約値ABR3を磁気ディスク装置17のコンテンツファイル21に記録する。更に、情報処理装置10は、コンテンツファイル21中の情報を通信ネットワーク1を介してコンテンツのプロバイダ3に送信する。これによって、コンテンツファイル21中の情報はプロバイダ3のサーバ内の閲覧可能なコンテンツファイル31に記録されて、第3
15 者がインターネットを介して自由に閲覧できるようになる。この結果、第3者は、公開されている試料Gの配列の数NA3、及び要約値AB3（又は必要に応じてABR3）を自分の保有するアミノ酸の配列の配列数、及び要約値と比較することによって、その試料Gが自分にとって新規かどうかを判定できる。また、ユーザは、その試料Gの配列情報を複数の供給者から誤って重複して購
20 入することを回避することができる。

その後、コンピュータシステム2Aの所有者（アミノ酸情報の供給者）は、ユーザから購入要求が来るのを待つ状態となる。そして、ユーザから試料Gに対する簡易データの要求があったときには、情報処理装置10は、磁気ディスク装置17のワーキングファイル20の中の試料Gのシンドロームの情報を例
25 えば電子メールの添付ファイルとしてそのユーザに送信する。シンドロームの情報を購入したユーザは、試料Gと同じ種類の自分で解読した試料のアミノ酸

の配列のシンδροームと、その購入したシンδροームとを比較することによって、相違する部分の検出及び復元を或る程度行うことができる。

一方、ユーザから完全データの要求があったときには、情報処理装置 10 は、ワーキングファイル 20 中のテキストデータ TX 3 を ZIP ファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値 AB 3 を同時に送信してもよい。本例によれば、簡易データ（シンδροーム）はデータ量が少ないために短時間で送信することができる。

更に、そのアミノ酸の配列情報の供給者は、ワーキングファイル 20 に記録した情報、即ち試料 G の名前の情報、配列の数 NA 3、テキストデータ TX 3、要約値 AB 3、ABR 3、及びシンδροームを CD-R/RW ドライブ 15 を介して CD-R 16 に記録してもよい。この CD-R 16 から、更に多数の CD-ROM を作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

次に、本例において、アミノ酸の配列中から所望の連続する部分的な配列を選択する方法の一例につき説明する。そのため、図 13 の試料 G の配列が、図 1 の表示装置 12 の表示画面中に表示されているものとして、その表示画面の右端のエッジ部を図 13 のエッジ部 51 とする。

図 13 において、試料 G のアミノ酸の配列はエッジ部 51 の左側の表示領域に表示されており、その表示領域には図 1 のマウス 204 によって制御されるカーソル 52 も表示されている。この場合、図 13 の第 16 列の第 2 行～第 7 行の矩形枠で囲まれた領域 54 内の配列を選択するものとする、本例では先ず領域 54 の右端部の文字 "A" の上にカーソル 52 を移動して、図 1 のマウス 204 の左スイッチ 204a を操作する。その後、カーソル 52 がエッジ部 51 から更に右方向の位置 53 まで仮想的に移動するように、マウス 204 を右方向に移動する。

本例では、そのようにカーソル 5 2 が一方のエッジ部に達した後にも、更にカーソル 5 2 が表示領域の外側に移動するようにマウス 2 0 4 を移動すると、そのカーソル 5 2 は、そのエッジ部に対向する他方のエッジ部からその表示領域内に現れるというスクリーン・ラッピング動作が行われる。この結果、カーソル 5 2 は、図 1 3 の表示領域の不図示の左側のエッジ部の右側に移動して、領域 5 4 の左端部の文字” K ” 上に移動して、領域 5 4 の配列が選択される。この状態で一例としてマウス 2 0 4 の右スイッチ 2 0 4 b を操作することによって、領域 5 4 の配列のコピー等を行うことができる。

次に、図 1 4 は、図 1 3 の配列の第 1 5 列～第 1 7 列の配列を示し、この図 1 4 において、第 1 6 列の第 8 行の領域 5 6 A、及びこれに続く第 1 7 列の第 1 行の領域 5 6 B の配列を選択するものとする。このとき、先ず領域 5 6 A の左端部の文字” L ” の上にカーソル 5 2 を移動して、図 1 のマウス 2 0 4 の左スイッチ 2 0 4 a を操作する。その後、カーソル 5 2 がエッジ部 5 1 から更に右下方向の位置 5 5 まで仮想的に移動するように、マウス 2 0 4 を右下方向に移動する。この結果、スクリーン・ラッピング動作によって、本例のカーソル 5 2 は、図 1 4 の表示領域の不図示の左側のエッジ部の右側に移動して、領域 5 6 B の右端部の文字” L ” 上に移動して、領域 5 6 A、5 6 B の配列が選択される。この状態で一例としてマウス 2 0 4 の右スイッチ 2 0 4 b を操作することによって、領域 5 6 A、5 6 B の配列のコピー等を行うことができる。

このように本例によれば、カーソルのスクリーン・ラッピング動作によって、マウス 2 0 4 の移動量を少なくしてアミノ酸の配列中の一連の広い領域、及び左右に離れた端部の連続する領域の配列を容易に選択することができる。同様に、ヌクレオチドの配列中から所定の部分的な領域を選択する場合にも、カーソルのスクリーン・ラッピング動作を行うことによって、選択動作を容易にかつ高速に行うことができる。

次に、カーソルのスクリーン・ラッピング動作の別の例につき図 1 7 を参照

して説明する。この例では、ユーザが図1のマウス204を用いて所望のアプリケーション・プログラムを起動する際の動作につき説明する。ここでは、図1の表示装置12の表示領域を図17の表示領域201aとして、表示領域201aの長辺方向をx方向、短辺方向をy方向とする。また、カーソルの座標を指定できる範囲を有効座標領域201bとする。この場合、カーソルの座標を表示領域201aの外側で、かつ有効座標領域201bの内側に設定すると、カーソルは表示領域201aのエッジ部に表示される。

図17(a)は、表示領域201aに表示されるプログラムリストの一例を示し、この図17(a)の表示領域201aには、メニューリスト221から選択されたプログラムの第1のグループリスト222(第1列)、及び第2のグループリスト223(第2列)がx方向に2列に分けて表示されている。この表示は、メニューリスト221上で「プログラム」の表示(反転している)上をカーソル220が通過することによって生成される。本例では、第2のグループリスト223中のグループG16中の或るアプリケーション・プログラムを実行したいものとして、カーソル220を「グループG16」の表示(反転している)上に移動させる。本例の表示領域201aでは、第2のグループリスト223(第2列)の右側(+x方向)にはサブ情報を表示する余地が無いため、グループG16のアプリケーションリスト224は、グループリスト222(第1列)の左側(-x方向)に表示される。ここで、実行したいアプリケーション・プログラムがアプリケーションA3であるとする、どのようにカーソル220をアプリケーションリスト224上に移動するかが問題となる。

即ち、単にカーソル220をグループG16上から左側のグループリスト222上に移動すると、例えばグループG2のアプリケーションリストが表示され、グループG16のアプリケーションリスト224の表示が消えてしまう。そこで、本例では、カーソル220をグループG16上から右方向(+x方向)

に移動させる。そして、カーソル220の座標を $P(m, n)$ とすると、更にカーソル220の座標が、表示領域201aを囲む有効座標領域201bの+x方向の外側の座標 $P(m1, n1)$ となるように、図1のマウス204を右方向に移動する。

- 5 この結果、カーソル220は、グループG16の表示の右側の位置から、図17(b)に示すように座標 $P(0, n1)$ の位置の近傍、即ちアプリケーションリスト224上に移動する。この後、マウス204を僅かに下方向に移動して、カーソル220をアプリケーションA3の表示上に移動させた状態で、図1の左スイッチ204aをクリックすることによって、極めて短時間に、かつ容易にアプリケーションA3のプログラムを起動することができる。

- 10 次に、本例では、例えば図17(a)において、カーソル220の計算上の座標が表示領域201aの外部で、かつ有効座標領域201bの内部にある（カーソル220は表示領域201aのエッジ部に表示されている）とき、即ちカーソル220が不活性である（アイドリング状態にある）ときには、図1
15 のマウス204のスイッチ204a, 204bに別の機能を持たせるようにしてもよい。このようにスイッチ204a, 204bに別の機能を持たせるときには、カーソル220の形状を変形させてもよい。一例として、そのようにカーソル220が不活性であるときに、左スイッチ204aを操作しながらマウス204をドラッグしたときには、表示領域201aの大きさを所定範囲で伸
20 縮できるようにしてもよい。更に、上記の範囲のみならず、例えばカーソル220が表示領域201aの輪郭（エッジ部）に対して内側に隣接する幅L1の棒状の領域にあるときにも、スイッチ204a, 204bに対して別の機能を持たせてもよい。

- 25 以上をまとめると、本例による情報選択方法は、複数の情報（221～223）が表示された表示領域（201a）より、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択する情報選択方法において、移

動量及び移動方向の少なくとも一方の情報を含む制御情報を生成し、この生成された制御情報に基づいてその表示領域内にその複数の情報に重畳させて移動自在にカーソル（２２０）（ポインタ）を表示し、このカーソルとその複数の情報の表示との位置関係に基づいて、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択できるようにしておき、そのカーソルをその表示領域の周縁部の第１の端部に移動させた状態で、更にそのカーソルをその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソルをその表示領域のその周縁部のその第１の端部とは異なる第２の端部を起点としてその表示領域内で移動させるものである。

即ち、本例のカーソル（２２０）は、ポインティングデバイスの制御情報に応じて、スクリーン・ラッピング方式で表示領域（２０１ａ）中を周期的に移動する。

この結果、ＧＵＩ (Graphical User Interface) 方式でコンピュータ等の各種装置を操作する際に、登録してあるアプリケーション・プログラムの個数が多い場合でも、高速にカーソルを所望のアプリケーション・プログラムの位置に移動させて、そのプログラムを起動することができる。

本例において、その表示領域（２０１ａ）が所定の軸に関して実質的に軸対称の領域（矩形領域、又は楕円形の領域等）である場合、その第２の端部は、その表示領域内でその所定の軸に関してその第１の端部と実質的に軸対称の位置に設定されると共に、そのカーソル（２２０）のその第２の端部からの移動方向は、その制御情報によってその第１の端部から更にそのカーソルを移動させようとした方向であることが望ましい。これによって、カーソル（２２０）の周期的な動きが容易に予測できるため、ユーザが特に習熟することなく、直ぐにその周期的なカーソル（２２０）の動きを活用できる。

また、そのカーソル（２２０）をその表示領域（２０１ａ）のその周縁部の第１の幅の制限領域に移動させた状態、及びそのカーソルをその制限領域から

更に第2の幅(L)だけ外側(201b)に移動させるようにその制御情報を与えた状態では、そのカーソルに対してその情報の選択以外の別の機能を与えることが望ましい。通常は、その表示領域(201a)の周縁部にはアプリケーション・プログラムのアイコン等は表示されていない。そこで、そのカーソル(220)がその表示領域(201a)の周縁部に有る状態では、アプリケーション・プログラムの選択以外の機能、例えばその表示領域の伸縮機能等を持たせても、アプリケーション・プログラムの選択には実質的に影響が無いと共に、カーソル(220)(ポインティングデバイス)の用途が広がる利点がある。

また、本例では図17に示すように、その表示領域(201a)内にその複数の情報、及びこれらの情報に関連する情報が複数列(222, 223)に表示されているときに、そのカーソル(220)がその複数列の一方の端部の列(223)の所定の情報の表示を通過しているときに、その表示領域内のその複数列の他方の端部の列(222)の外側にその所定の情報に関連する複数のサブ情報(224)を表示し、更にそのカーソルをその複数列の一方の端部の列(223)からその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソル(220)をその表示領域のその複数列の他方の端部の列(222)に近い端部P(0, n1)を起点として、その複数のサブ情報(224)の表示上に移動させて、その複数のサブ情報の何れかを選択可能としている。

即ち、図17に示すように、その表示領域(201a)内に表示すべきアプリケーション・プログラムの個数が多い場合には、例えばその右側の端部の列(223)のサブ情報(224)が左側の端部の列(222)の外側に表示される。このときに、本例の周期的な移動を行うカーソル(220)を適用すると、そのサブ情報(224)中のアプリケーション・プログラムを選択するためには、ポインティングデバイスによってそのカーソル(220)をその列

(2 2 3) から更に右方向に移動させるようにすればよい。これによって、アプリケーション・プログラムの個数が多く、プログラムリストが複数列になるような場合でも、GUI方式で容易にカーソルを所望のアプリケーション・プログラムの位置に移動できる。

5 ここで、DNA又はRNAのヌクレオチドの配列（塩基の配列）に対応するテキストデータ（又はこれを表1等に基づいて変換した数値データ）の要約値を算出するためのハッシュ関数について更に説明する。例えばハッシュ関数の演算対象を、人間のDNAのヌクレオチドの配列とすると、そのテキストデータ又は数値データのファイル（以下、「原ファイル」と言う）は100Mバイト程度にも達する膨大なファイルである。そこで、本発明で使用するハッシュ関数（ハッシュ演算アルゴリズム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

10

 また、ハッシュ関数は、一例として所定ビット数 m_1 （ m_1 は例えば32, 64等）のデータを1ワードとして、所定ワード数 m_2 （ m_2 は例えば16, 32, 64等）単位で、原ファイルの要約値を算出していく。この際に、データの処理単位は、 $m_1 \cdot m_2$ ビットとなる。例えば $m_1 = 32$, $m_2 = 16$ では、処理単位は512ビットとなる。そこで、その原ファイルを複数の分割ファイルに分割する際には、最初は $m_1 \cdot m_2$ ビットの整数倍（例えば10000倍程度）を単位として分割していき、端数として残ったデータに所定データ（長さを表すデータ、区切りデータ等）を付加して $m_1 \cdot m_2$ ビットの整数倍のファイルとすることで、要約値の演算を効率的に実行することができる。

15

20

 更に、暗号理論で使用されるハッシュ関数は、テキストデータ中のスペースコード及び改行コード等も全て演算処理対象としているが、ヌクレオチド及びアミノ酸の配列情報については見やすくするために、配列番号1～3で示すように途中にスペースコード、配列順序を示す数字コード、及び改行コードを挿

25

入する場合がある。そこで、ヌクレオチドの配列情報（アミノ酸の配列情報も同様）を演算処理対象とするハッシュ関数においては、必要に応じてテキストデータ中の所定コードとしての数字コード、スペースコード及び改行コードを無視する機能を付加することが望ましい。また、隣接する文字を” - ”（ハイフン）で分けることも考えられるが、この場合には、更に” - ”記号も無視する必要がある。

更に、原ファイルを複数の分割ファイルに分割する際には、複数の分割ファイルの順序等を示すデータ（以下、「コメントデータ」と言う）を各分割ファイルに付加することが望ましいことがある。このように分割ファイル、又は1つの原ファイルにコメントデータを付加する場合にも、コメントデータはハッシュ関数で無視する必要がある。そのため、例えばコメントデータは所定の開始記号（例えば / * ）及び終了記号（例えば * / ）の間に記録し、ハッシュ関数で処理する際に開始記号から終了記号までのデータは無視するようにすればよい。

また、上記の実施の形態では、例えば生物のDNAのヌクレオチドの配列（又はタンパク質のアミノ酸の配列）内の先頭の一部、及び末尾の一部の配列、並びにその配列のテキストデータの要約値をインターネット上で公開することがある。この場合には、その公開されている一部の配列と、その要約値とからそのテキストデータの内容が推定される可能性もある。これを回避するために、そのテキストデータをハッシュ関数で処理する際に、その公開されている配列を除いた部分についてのみ、そのハッシュ関数を施して要約値を求めるようにしてもよい。

次に、例えば核酸や遺伝子のヌクレオチドの配列が見易いように順序を示す数字、スペース、及び改行を含んでテキストデータとして記録されたファイル（ファイルFD1とする）の要約値（message digest）を計算するための方法の一例につき図15を参照して説明する。なお、以下の要約値の計算は、例えば

図1の情報処理装置10において実行される。

図15において、先ずステップ151では、ファイルFD1中のテキストデータから数字コード、スペースコード、及び改行コードを取り除いたテキストデータをファイルFD2に記録する。その次のステップ152では、例えばMD5ハッシュ関数を用いてファイルFD2中のテキストデータの128ビットの要約値を算出する。この方法は処理は単純であるが、ファイルFD1が例えば100Mバイト程度であるとする、ファイルFD2もほぼ100Mバイト程度になるため、記憶装置の容量を大きくする必要がある。

MD5ハッシュ関数のアルゴリズムについては、上記のウェブサイト2で詳細に開示されているが、ここでそのアルゴリズムについて簡単に説明する。

先ず、ここでは1ワード"word"とは、32ビットの量であり、1バイト"byte"とは、8ビットの量である。そして、一列のビットは、自然に一列のバイトと解釈することができ、ここではそれぞれ8ビットのデータの集まりを、MSB (most significant bit)、即ち上位ビットが最初に表示される1バイトのデータとして解釈することができる。同様に、一列のバイトは、一列の32ビットのワードと解釈することができ、ここではそれぞれ4バイトのデータの集まりを、LSB (least significant byte)、即ち下位バイトが最初に表示される1ワードのデータとして解釈することができる。

また、次のように演算等を定義する。即ち、" x_i "は x に下付き文字 i を付加した表現を意味し、その下付き文字が一つの式であるときには、例えば" $x_{|i+1|}$ "のようにその式を括弧で囲むものとする。同様に、上付き文字(べき乗)としては $^$ を用いる。従って、" x^i "は x の i 乗を意味する。

また、記号" $+$ "は、ワードの加算、即ち法 2^{32} の加算を意味する。そして、" $X \lll s$ "は、 X を s ビットだけ左側に循環的にシフトして(回転して)得られる32ビットの値を意味する。また、 $\text{not}(X)$ は、 X のビット毎の補数(complement)を意味し、" $X \vee Y$ "は、 X と Y とにビット毎のOR演算を施して得

られる値を意味し、"X xor Y"はX とY とにビット毎のXOR（排他的論理和）演算を施して得られる値を意味し、"XY"はX とY とにビット毎のAND演算を施して得られる値を意味する。

次に、上記のファイルFD2に記録されているテキストデータ（ファイルFD1から数字コード、スペースコード、及び改行コードを取り除いたテキストデータ）を、要約値を求めるべきbビットのメッセージであるとする。その値bは、任意の非負整数であり、bは0であってもよい。その値bは8の倍数である必要はなく、更に任意に大きい値であってもよい。そのbビットのメッセージの一連のビットは次のように表すことができる。

5 $m_0 \ m_1 \ \dots \ m_{[b-1]}$

そのメッセージの要約値は、次の5つのステップA～Eの処理で計算することができる。

〔ステップA〕（追加ビットの付加）

そのメッセージには、そのメッセージをビット列で表現したときの長さが法512のもとで448に合同となるように追加ビットが付加（拡張）される。即ち、そのメッセージは、その長さが512の倍数のビットよりも64ビットだけ少ない長さになるように拡張される。追加ビットの付加は、たとえそのメッセージの長さが既に法512のもとで448に合同である場合でも常に実行される。

20 追加ビットの付加は、単一のビット"1"を付加した後に、ビットの長さが法512のもとで448に合同となるようにビット"0"を付加することによって実行される。全ての場合に、少なくとも1ビット、そして最大で512ビットが付加される。

〔ステップB〕（長さ情報の付加）

25 そのメッセージのビット数であるb（ステップAにおける追加ビットの付加が行われる前の長さ）の64ビットの表現が、ステップAで得られたメッセー

ジに付加される。実際には起こりそうもないが、仮に b が 64 ビットよりも大きいときには、 b の表現の下位の 64 ビット分だけが付加される。これらのビットは、2つの 32 ビットのワードとして、上述の内容に対応して下位のワードが最初になるように付加される。

5 このようにして得られたメッセージのビット表現の長さは、正確に 512 の倍数、即ち 512 ビットの倍数となる。言い換えると、このようにして得られたメッセージの長さは、正確に 16 個の (32 ビットの) ワードの倍数となる。そこで、このようにして得られたメッセージの各ワードを $M[0 \dots N-1]$ とする。ここで、 N は 16 の倍数である。

10 [ステップ C] (要約値バッファの初期化)

要約値を計算するために 4 ワードのバッファ (A, B, C, D) を使用する。ここで、A, B, C, D はそれぞれ 32 ビットのレジスタであり、これらのレジスタは下位バイトを最初に記載する 16 進表現で次の値に初期化される。

ワード A: 01 23 45 67

15 ワード B: 89 ab cd ef

ワード C: fe dc ba 98

ワード D: 76 54 32 10

[ステップ D] 16 ワードブロック毎のメッセージの処理

20 ここでは、それぞれ入力として 3 個の 32 ビットのワードを受け取って出力として 1 個の 32 ビットのワードを生成する 4 個の補助的な関数を次のように定義する。

$$F(X, Y, Z) = XY \vee \text{not}(X) Z$$

$$G(X, Y, Z) = XZ \vee Y \text{ not}(Z)$$

$$H(X, Y, Z) = X \text{ xor } Y \text{ xor } Z$$

25 $I(X, Y, Z) = Y \text{ xor } (X \vee \text{not}(Z))$

各ビット位置で、関数 F は、 X が真ならば Y で、そうでなければ Z という条

件式として作用する。関数Fは、 v の代わりに $+$ を使って定義することもできた。なぜなら、 XY と $\text{not}(X)Z$ とは、同じビット位置で共に1となることが決してないからである。

関数G, H, Iは、 X, Y, Z のビットからビット毎に並行に出力を生成する点
5 で関数Fと同様である。

また、関数Hは、その入力に対してビット毎のXOR演算、又はパリティ演算を施す関数である。

更にこのステップDでは、正弦関数から導かれる64個の要素を持つテーブル $T[1 \dots 64]$ を用いる。即ち、そのテーブルの i 番目の要素を $T[i]$ として、
10 i の単位をラジアンとすると、次のようになる。

$$T[i] = \{4294967296 \times \text{abs}(\sin(i))\} \text{ の整数部}$$

なお、 $\text{abs}(\sin(i))$ は $\sin(i)$ の絶対値である。これらの関数及びテーブルを用いて以下の演算を行う。

各16ワードのブロックを処理するために、変数 i について0から $(N/16 - 1)$ まで以下の「 i に関するループの始まり」から「 i に関するループの
15 終わり」までの処理を繰り返して行う。

「 i に関するループの始まり」

先ず変数 j について0から15まで、1ワードのメッセージ $M[i*16+j]$ を $X[j]$ にコピーする。

20 続いて、バッファA, B, C, D の値をそれぞれ次のようにバッファAA, BB, CC, DD にコピーする。

$$AA = A, BB = B, CC = C, DD = D$$

「ラウンド1」

ここで、 $[abcd \ k \ s \ i]$ は次の処理を行うものと定義する。

$$25 \quad a = b + ((a + F(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の16回の処理を行う。

[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
 [ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
 [ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
 [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

5 [ラウンド 2]

ここで、[abcd k s i] は次の処理を行うものと定義する。

$$a = b + ((a + G(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 16 回の処理を行う。

10 [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
 [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
 [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
 [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

[ラウンド 3]

ここで、[abcd k s t] は次の処理を行うものと定義する。

15
$$a = b + ((a + H(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 16 回の処理を行う。

20 [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
 [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
 [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
 [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]

[ラウンド 4]

ここで、[abcd k s t] は次の処理を行うものと定義する。

$$a = b + ((a + I(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 16 回の処理を行う。

25 [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
 [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]

[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]

[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

次に、バッファA, B, C, D の値にそれぞれ次のようにバッファAA, BB, CC, DD の値（このブロックの処理が始まる前のバッファA, B, C, D の値）を加算する。

5 $A = A + AA, B = B + BB, C = C + CC, D = D + DD$

 [i に関するループの終わり]

 [ステップE] 出力

 出力として計算された要約値はバッファA, B, C, D の値そのものである。

10 即ち、バッファA の下位バイトから始まって、バッファD の上位バイトで終わる値がその要約値である。なお、要約値が32ビット又は64ビットでよいような場合には、それぞれ例えばバッファA、又はバッファA, B の値のみを要約値として用いてもよい。

15 また、MD5ハッシュ関数は、元のデータの推定が困難となるように複雑な処理を行っているが、ヌクレオチドやアミノ酸の配列データの要約値を計算する場合には元のデータが或る程度推定されても特に不都合がないことがある。この場合には、メッセージに対応する一連の所定ビット数 s (s はMD5ハッシュ関数では512) のブロック B_i ($i = 1 \sim I$) 毎の演算を、順次次のような簡単な演算で行うことも考えられる。

$$M_1 = (a \cdot B_1 + b) \bmod 2^s$$

20 $M_i = (M_{i-1} \cdot B_i + b) \bmod 2^s \quad (i = 2 \sim I)$

 この場合、 a , b は0以外の s ビットの数であり、 M_i が最終的な要約値となる。

25 次に、上記のファイルFD1の要約値を計算するための別の方法につき図16を参照して説明する。ここではMD5ハッシュ関数を用いて要約値を計算するものとする。以下の計算も一例として図1の情報処理装置10で実行される。

 先ず図16のステップ161において、ヌクレオチド自体を表すコードの個

数を表す変数NX, NYの値をそれぞれ0に設定し、要約値を表す32ビットずつのバッファA, B, C, Dの値を所定の初期値（上記のステップCで設定した値）に設定し、要約値の計算対象のテキストデータを空にする。

- 5 次のステップ162において、ファイルFD1中のテキストデータの先頭から1文字分（ここでは1バイト）の文字コード（ここでは全ての種類のコードを含む意味である）を読み取り、それに続くステップ163において、読み取った文字コードが数字コード、スペースコード、又は改行コードかをチェックする。そして、読み取った文字コードが数字、スペース、改行の何れのコードでもない、即ち本例ではA～Z, a～zの何れかのコードであるときには、ステップ164に移行して、変数NXの値に1を加算すると共に、読み取った文字コードを要約値の計算対象のテキストデータに加える。続いてステップ165において、変数NX（読み取られた有効な文字コードの個数）が、要約値の計算単位である文字数NAに達したかどうかを調べる。本例では、 $NA = 512 / 8 = 64$ である。
- 10 NX=NAであるときには、ステップ166に移行して、変数NXを0に戻すと共に、変数NY（NA個の文字単位のブロック数）に1を加算した後、ステップ167に移行して、計算対象のテキストデータ（NA個の文字コードを含んでいる）の要約値（A, B, C, D）を計算する。これは、上記のステップDを1回実行することを意味する。その後、計算対象のテキストデータを空
- 20 にしてから、動作はステップ168に移行して、ファイルFD1中に読み取り対象となる文字コードがまた有るかどうかチェックされる。また、ステップ163で読み取られた文字コードが数字、スペース、改行の何れかのコードであるときには、ステップ169で読み取った文字コードを無視した後、ステップ168に移行する。更に、ステップ165で変数NXがNAに達していない
- 25 ときにも、動作はステップ168に移行する。

そして、ステップ168において、読み取り対象となる文字コードがまだ有

るときには、動作はステップ162に戻り、ファイルFD1中のテキストデータから次の1文字分の文字コードが読み取られて、以下ステップ163～168の動作が繰り返される。一方、ステップ168において、読み取り対象となる文字コードが無くなったときには、動作はステップ170に移行して、要約値(A, B, C, D)が計算される。この際に、変数NX, NYの値より読み取られた有効な文字コードの全個数が分かるため、上記のステップA、ステップB、ステップD、ステップEが実行される。得られた要約値(A, B, C, D)が最終的な要約値となる。

具体的にMD5ハッシュ関数を用いて図15、及び図16の計算方法で、配列番号1、2のヌクレオチドの配列を示すテキストデータ、及び配列番号3のアミノ酸の配列を示すテキストデータの要約値を計算した結果は、16進数表示で以下のようなになる。これらの要約値は、配列の改行方法等を変えても変化しない一定の値である。

MD5の要約値(配列番号1) = hex(1c0a0b1d72e256bb10556a2fb52d28ae)
MD5の要約値(配列番号2) = hex(ec8c3c9af5630f61f3d0cd2bd13b0f0d)
MD5の要約値(配列番号3) = hex(164f14406ac21158e20ba72666a033ab)

この要約値の計算方法によれば、ファイルFD1から逐次関係の無い文字コードを取り除きながら要約値を計算しているため、記憶装置の記憶容量を殆ど増加する必要がないという利点がある。従って、ファイルFD1の情報量が大きくなる程、この計算方法は有利になる。また、上記のステップ151、163では、所定コードとしての数字コード、スペースコード、改行コードを取り除いているが、それ以外にコメント文などを取り除くようにしてもよい。また、図15、図16で要約値を計算するための方法は、MD5ハッシュ関数のみならず、他のどのような関数であってもよい。

なお、本発明は上述の実施の形態に限定されず、本発明の要旨を逸脱しない範囲で種々の構成を取り得ることは勿論である。また、明細書、特許請求の範

囲、図面、及び要約を含む2000年4月19日付け提出の日本国特願2000-117343の開示内容の要部、及び2000年5月19日付け提出の日本国特願2000-149122の開示内容の全ては本願に組み込まれている。

5 産業上の利用の可能性

本発明によれば、核酸や遺伝子中のヌクレオチドの配列情報、又はタンパク質やペプチド中のアミノ酸の配列情報を、それらの配列が所定の長さを超えたときに、それらの配列を示すテキストデータよりも少ないデータ量で記録することができる。従って、それらの配列情報を通信回線を介して短時間に送信することが可能となる。

また、それらの配列を示すテキストデータ、又はこれに対応する数値データの数学的な要約値を用いた場合には、膨大な長さの2つのヌクレオチドの配列同士、又は2つのアミノ酸の配列同士の同一性を少ないデータ量で高精度に確認することができる。また、同一の複数の配列情報を誤って購入することも防止できる。

また、シンδροーム情報を用いた場合には、2つのヌクレオチドの配列（又は2つのアミノ酸の配列）の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元することができる。従って、例えばSNP（一塩基変位多型：Single Nucleotide Polymorphism）を少ないデータ量で容易に発見することができる。

また、本発明によれば、ヌクレオチドの配列情報、又はアミノ酸の配列情報を少ないデータ量でユーザに供給できるビジネスモデルを提供することができる。この場合に、更に数学的な要約値、又はシンδροーム情報を用いることによって、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性の確認、又は相違する部分の検出や復元を容易に行うことができる。

また、本発明の要約値の計算方法によれば、例えばヌクレオチドの配列を見

易くするための数値コードやスペースコードなど、又はその配列の内容を説明するためのコメント文などの所定コードを無視して、必要な情報のみの要約値を算出できるため、その所定コードの内容が変化しても、常に同一の要約値を算出できる利点がある。従って、その要約値の計算方法は、特にヌクレオチド
5 やアミノ酸の配列情報の要約値を算出する場合に有効である。

10

15

20

25

請 求 の 範 囲

1. 一列のヌクレオチドの配列情報の記録方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報を記録することを特徴とするヌクレオチドの配列情報の記録方法。

2. 請求の範囲 1 記載の記録方法であって、

前記一列のヌクレオチドは 4 種類のヌクレオチドよりなり、

前記 4 種類のヌクレオチドを互いに異なる 6 ビット以下のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

3. 請求の範囲 2 記載の記録方法であって、

前記 4 種類のヌクレオチドを互いに異なる 2 ビットのデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

4. 請求の範囲 2、又は 3 記載の記録方法であって、

前記一列のヌクレオチドは、一つの DNA を構成する 1 対の重合体の鎖の内の 1 本の鎖の全部又は一部であり、

前記 4 種類のヌクレオチド中の互いに相補的な 2 対のヌクレオチドをそれぞれ互いにビット反転の関係にある 1 対のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

5. 請求の範囲 2、又は 3 記載の記録方法であって、

前記一列のヌクレオチドは、一つの RNA を構成する一つの重合体の鎖の全部又は一部であることを特徴とするヌクレオチドの配列情報の記録方法。

6. 請求の範囲 1 記載の記録方法であって、

前記一列のヌクレオチドの配列に関する情報を、前記配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

7. 請求の範囲 6 記載の記録方法であって、

前記一列のヌクレオチドは 25 個以上のヌクレオチドの配列であり、

前記一列のヌクレオチドの配列に関する情報を 40 ビット以上で 192 ビット以下の長さの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

8. 請求の範囲 7 記載の記録方法であって、

前記数学的な要約値は、前記一列のヌクレオチドの配列に対応するテキストデータ又は数値データに MD5 ハッシュ関数、又は SHA ハッシュ関数の演算を施して得られることを特徴とするヌクレオチドの配列情報の記録方法。

9. 請求の範囲 1 記載の記録方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報で前記一列のヌクレオチドの配列を表すことを特徴とするヌクレオチドの配列情報の記録方法。

10. 請求の範囲 9 記載の記録方法であって、

複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交互に第 1 群の変換データ及び第 2 群の変換データに分けたとき、

前記第 1 の演算は、所定の整数 K を用いて前記第 1 群の変換データ、及び前

記第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、

前記第 2 の演算は、複数列の前記変換データの各列の変換データに対する法 K のもとの和を求める演算であることを特徴とするヌクレオチドの配列情報の記録方法。

5 1 1. 請求の範囲 9、又は 10 記載の記録方法であって、

前記一列のヌクレオチドの配列を基準配列として、該基準配列の 2 組の前記シンドローム情報に対応させて、検査対象の一列のヌクレオチドの配列の 2 組のシンドローム情報を求め、

10 前記 4 組のシンドローム情報より前記基準配列に対する前記検査対象の一列のヌクレオチドの配列の相違部を求めることを特徴とするヌクレオチドの配列情報の記録方法。

1 2. 一列のヌクレオチドの配列情報の記録装置であって、

一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置と、

15 該配列読み取り装置で読み取られた配列の情報をテキストデータとして第 1 ファイルに記録する第 1 記録手段と、

前記第 1 ファイルのテキストデータよりも少ないデータ量で、前記配列読み取り装置で読み取られた配列の情報を表し、該配列の情報を第 2 ファイルに記録する第 2 記録手段と

20 を有することを特徴とするヌクレオチドの配列情報の記録装置。

1 3. 請求の範囲 12 記載の記録装置であって、

前記第 2 記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオチドの配列を、該配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録装置。

25 1 4. 請求の範囲 12 記載の記録装置であって、

前記第 2 記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオ

チドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

5 前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

10 複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報を前記第 2 ファイルに記録することを特徴とするヌクレオチドの配列情報の記録装置。

1 5. 一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、

15 前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報が記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

1 6. 請求の範囲 1 5 記載の記録媒体であって、

前記一列のヌクレオチドは 2 5 個以上のヌクレオチドの配列であり、

20 前記一列のヌクレオチドの配列に関する情報は、4 0 ビット以上で 1 9 2 ビット以下の長さの数学的な要約値で前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

1 7. 請求の範囲 1 5 記載の記録媒体であって、

25 前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し、

5 複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求めておき、

10 前記一列のヌクレオチドの配列に関する情報は、前記第 1 組及び第 2 組のシンδροーム情報として前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

1 8. 一列のヌクレオチドの配列情報の供給方法であって、

15 前記一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

前記一列のヌクレオチドの配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

20 前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするヌクレオチドの配列情報の供給方法。

25 1 9. 請求の範囲 1 8 記載の供給方法であって、

前記一列のヌクレオチドは 2 5 個以上のヌクレオチドの配列であり、

前記数学的な要約値は、40ビット以上で192ビット以下のデータであり、
前記供給者は、更に前記一列のヌクレオチドの所定の一部の配列の情報を通信回線を介して閲覧可能な状態にしておくことを特徴とするヌクレオチドの配列情報の供給方法。

5 20. 請求の範囲18、又は19記載の供給方法であって、

前記供給者は、前記一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応する前記数値データを第1ファイルに記録して保持し、

10 前記供給者は、前記テキストデータ、又は前記数値データを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、

15 複数列の前記変換データに各列毎に前記配列方向に第2の演算を施して第2組のシンδροーム情報を求め、

前記第1組及び第2組のシンδροーム情報を第2ファイルに記録して保持し、

第1段階として前記ユーザは、前記供給者より前記第2ファイルに記録されている2組のシンδροーム情報を受け取り、

20 前記2組のシンδροーム情報に基づいて検査対象の一列のヌクレオチドの配列の内の前記供給者の一列のヌクレオチドの配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第2段階として前記ユーザは前記供給者より前記第1ファイルに記録されている前記テキストデータ、又は前記数値データの内の前記配列の復元ができない部分の情報の提供を要求することを特徴とするヌクレオチドの配列情報の供給方法。

25 21. 一列のアミノ酸の配列情報の記録方法であって、

前記一列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、前記一列のアミノ酸の配列に関する情報を記録することを特徴とするアミノ酸の配列情報の記録方法。

2 2. 請求の範囲 2 1 記載の記録方法であって、

- 5 前記一列のアミノ酸は、一つのタンパク質を構成する 1 本のアミノ酸の鎖の全部又は一部であり、

前記一列のアミノ酸の配列に対応するテキストデータを、20 種類のアミノ酸に対して互いに異なる 6 ビット以下のデータを割り当てることによって変換することを特徴とするアミノ酸の配列情報の記録方法。

- 10 2 3. 請求の範囲 2 1 記載の記録方法であって、

前記一列のアミノ酸の配列に関する情報を、前記配列を表すテキストデータの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

2 4. 請求の範囲 2 3 記載の記録方法であって、

前記一列のアミノ酸は 2 5 個以上のアミノ酸の配列であり、

- 15 前記一列のアミノ酸の配列に関する情報を 1 6 ビット以上で 1 9 2 ビット以下の長さの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

2 5. 請求の範囲 2 3、又は 2 4 記載の記録方法であって、

- 20 前記数学的な要約値は、前記一列のアミノ酸の配列に対応するテキストデータに MD 5 ハッシュ関数、又は SHA ハッシュ関数の演算を施して得られることを特徴とするアミノ酸の配列情報の記録方法。

2 6. 請求の範囲 2 1 記載の記録方法であって、

- 25 前記一列のアミノ酸の配列に対応するテキストデータを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに異

なる 8 ビット以下の数値データを割り当てることによって変換データに変換し、
複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第
1 組のシンδροーム情報を求めると共に、

5 複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2
組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報で前記一列のアミノ酸の配列を表
すことを特徴とするアミノ酸の配列情報の記録方法。

27. 請求の範囲 26 記載の記録方法であって、

10 複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交
互に第 1 群の変換データ及び第 2 群の変換データに分けたとき、

前記第 1 の演算は、所定の整数 K を用いて前記第 1 群の変換データ、及び前
記第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、

15 前記第 2 の演算は、複数列の前記変換データの各列の変換データに対する法
 K のもとの和を求める演算であることを特徴とするアミノ酸の配列情報の記録
方法。

28. 一列のアミノ酸の配列情報の記録装置であって、

一つのタンパク質の少なくとも一部に含まれる一列のアミノ酸の配列情報を
テキストデータとして第 1 ファイルに記録する第 1 記録手段と、

20 前記第 1 ファイルのテキストデータよりも少ないデータ量で、前記一列のア
ミノ酸の配列の情報を表し、該配列の情報を第 2 ファイルに記録する第 2 記録
手段と

を有することを特徴とするアミノ酸の配列情報の記録装置。

29. 請求の範囲 28 記載の記録装置であって、

25 前記第 2 記録手段は、前記一列のアミノ酸の配列を、該配列を表すテキスト
データの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録装
置。

30. 一列のアミノ酸の配列情報の供給方法であって、

前記一列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる8ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

5 前記一列のアミノ酸の配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするアミノ酸の配列情報の供給方法。

31. 請求の範囲30記載の供給方法であって、

前記一列のアミノ酸は25個以上のアミノ酸の配列であり、

15 前記数学的な要約値は、16ビット以上で192ビット以下のデータであることを特徴とするアミノ酸の配列情報の供給方法。

32. 請求の範囲30、又は31記載の供給方法であって、

前記供給者は、前記一列のアミノ酸の配列に対応するテキストデータ、又はこれに対応する前記数値データを第1ファイルに記録して保持し、

20 前記供給者は、前記テキストデータ、又は前記数値データを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のアミノ酸に対して互いに異なる8ビット以下の数値データを割り当てることによって変換データに変換し、

25 複数行の前記変換データに各行毎に前記非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報を第 2 ファイルに記録して保持し、

第 1 段階として前記ユーザは、前記供給者より前記第 2 ファイルに記録されている 2 組のシンδροーム情報を受け取り、

前記 2 組のシンδροーム情報に基づいて検査対象の一系列のアミノ酸の配列の内の前記供給者の一系列のアミノ酸の配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第 2 段階として前記ユーザは前記供給者より前記第 1 ファイルに記録されている前記テキストデータ、又は前記数値データの情報の提供を前記供給者に要求することを特徴とするアミノ酸の配列情報の供給方法。

3 3. 一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、

前記一つ又は複数のファイルに記録されたデータの中で所定のコードを無視して要約値を計算することを特徴とする要約値の計算方法。

3 4. 請求の範囲 3 3 記載の要約値の計算方法であって、

前記無視する所定のコードは、数字コード、スペースコード、及び改行コードであることを特徴とする要約値の計算方法。

3 5. 請求の範囲 3 3 記載の要約値の計算方法であって、

前記無視する所定のコードは、同一又は互いに異なる 2 組のコード、及びこれら 2 組のコードに挟まれたデータであることを特徴とする要約値の計算方法。

3 6. 請求の範囲 3 3 記載の要約値の計算方法であって、

前記一つ又は複数のファイルから 1 文字分のコードデータを読み出す毎に、

該読み出されたコードデータが前記所定のコードであるときには、該読み出されたコードデータを無視して、次の 1 文字分のコードデータの読み出しを行い、

該読み出しによって得られた前記所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うことを特徴とする要約値の計算方法。

37. 一連のテキストデータの要約値を計算するための要約値の計算方法であって、

前記一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、前記所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、

前記複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、

該複数のファイルに記録されたテキストデータから分割の順序に従って順次要約値を計算することを特徴とする要約値の計算方法。

38. 請求の範囲37記載の要約値の計算方法であって、

前記所定個数ずつのコードデータ、及び前記所定個数よりも少ない個数のコードデータからは、所定のコードデータが除外されていることを特徴とする要約値の計算方法。

39. 請求の範囲38記載の要約値の計算方法であって、

前記所定個数は、要約値を計算する際のデータ量の単位に応じて定められることを特徴とする要約値の計算方法。

1/15

図 1

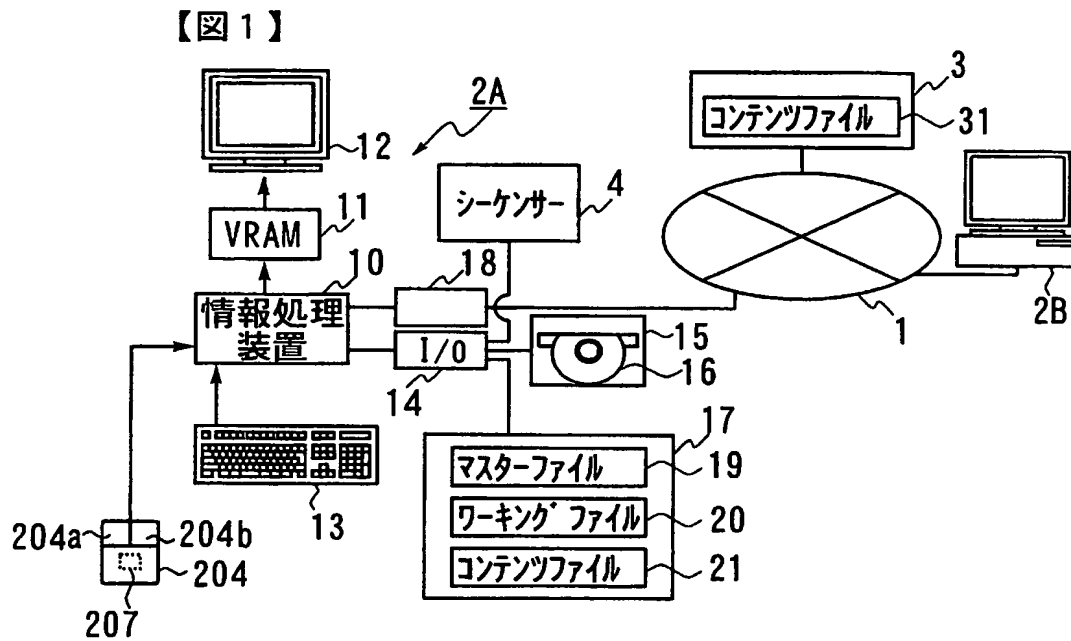
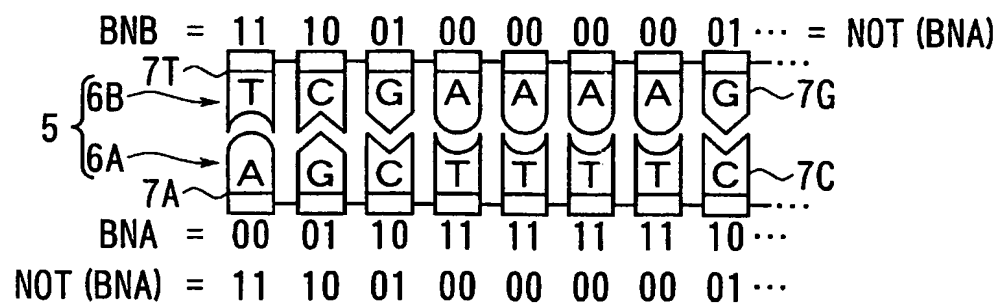
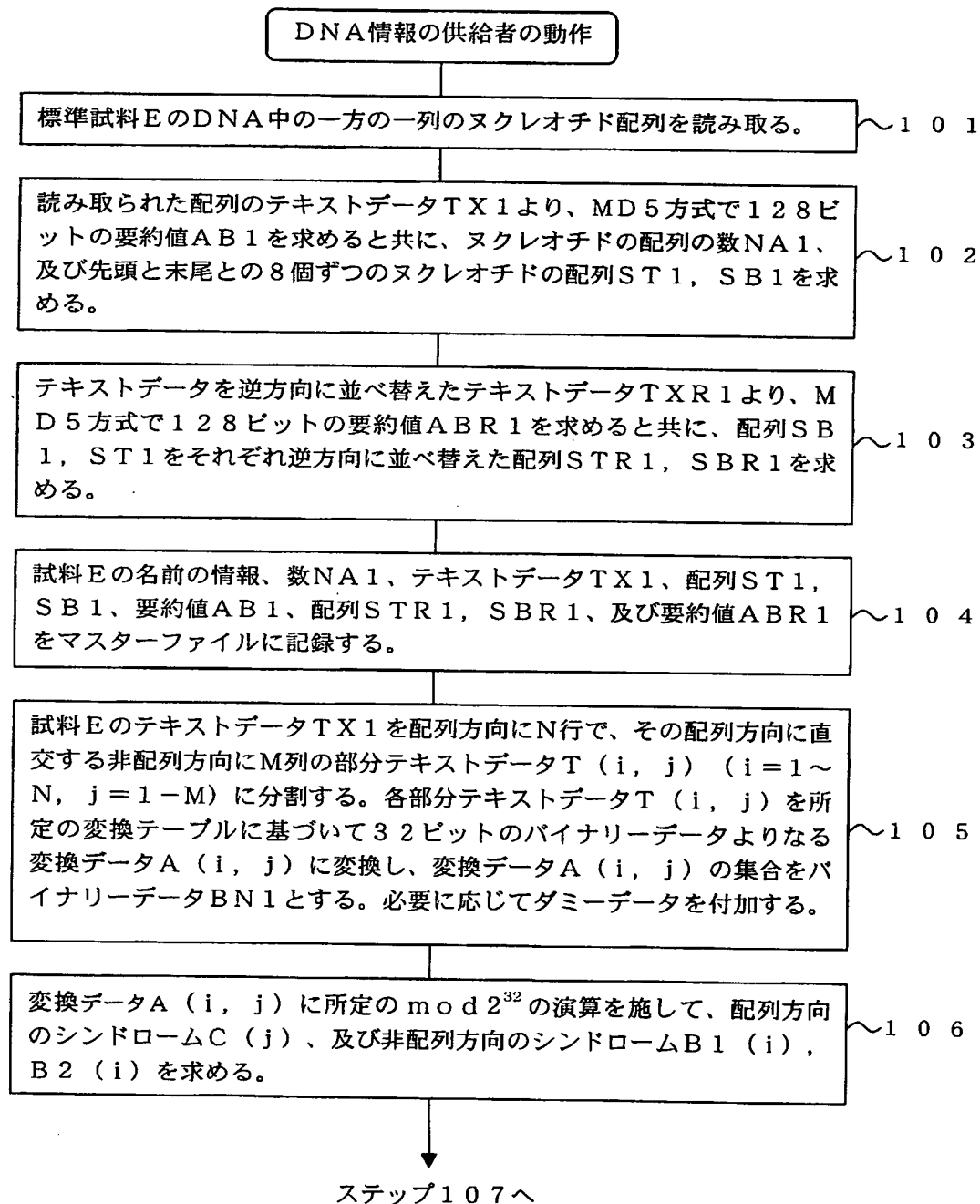


図 2



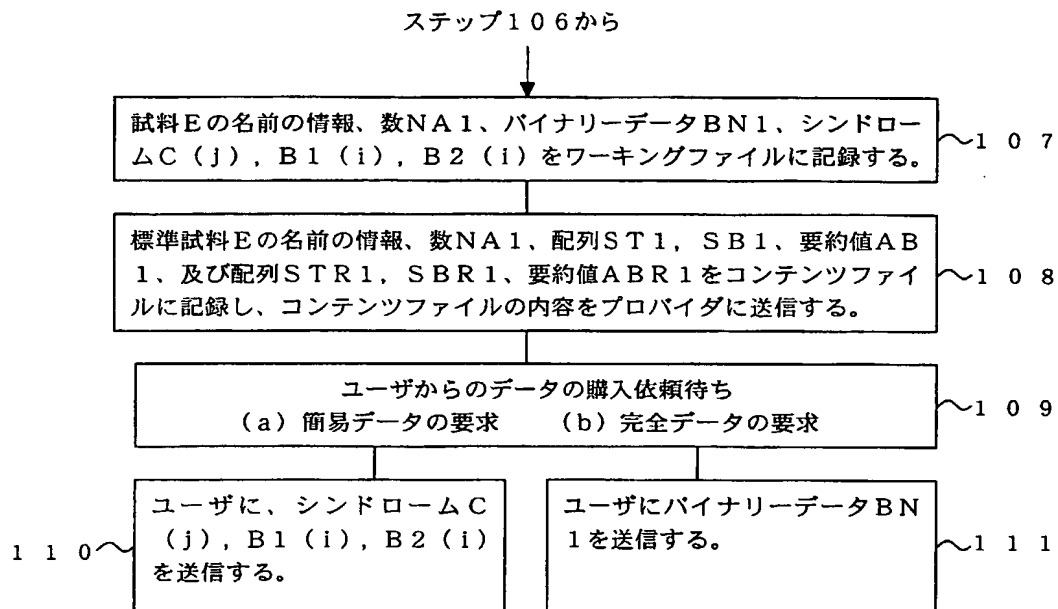
2/15

図 3



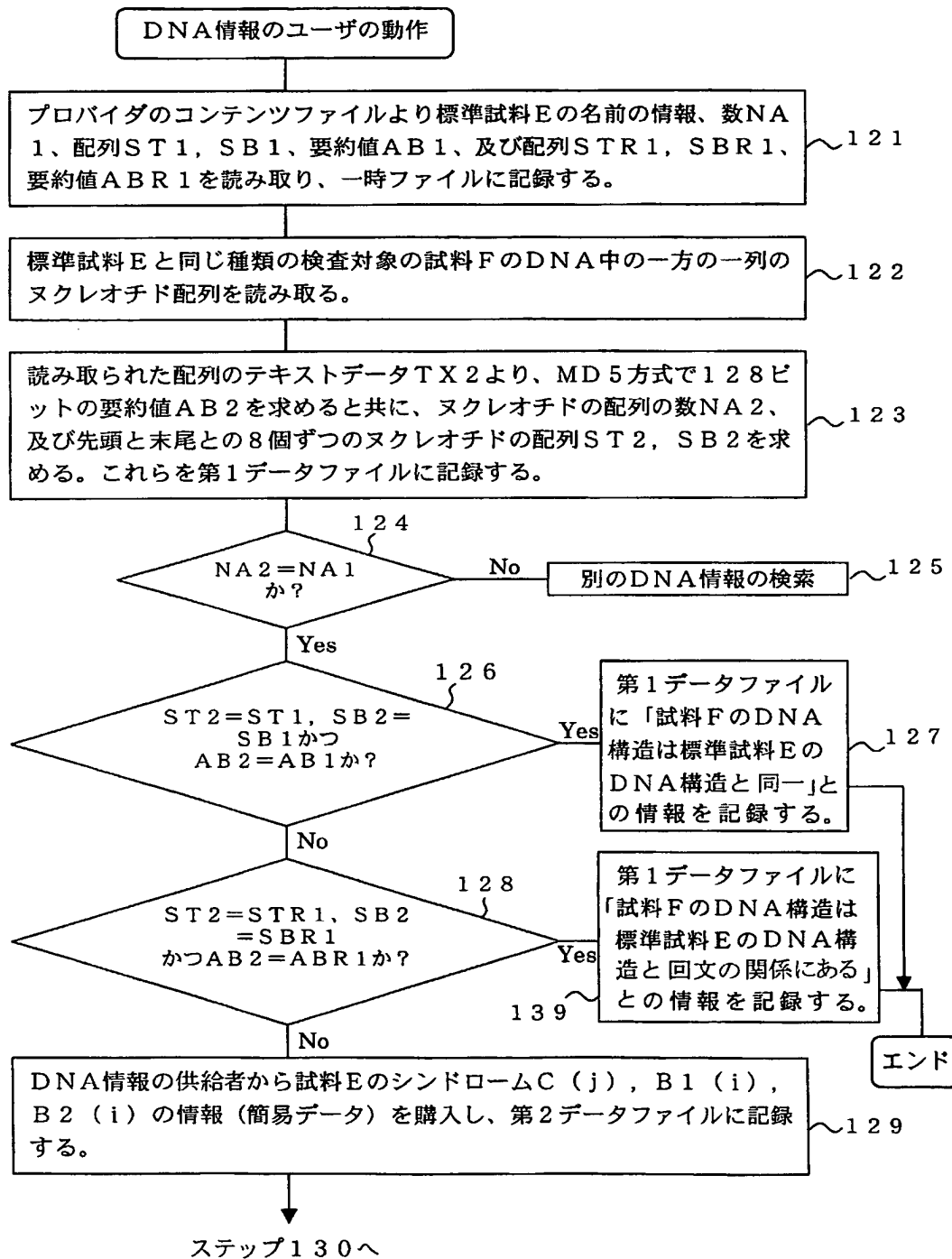
3/15

図 4



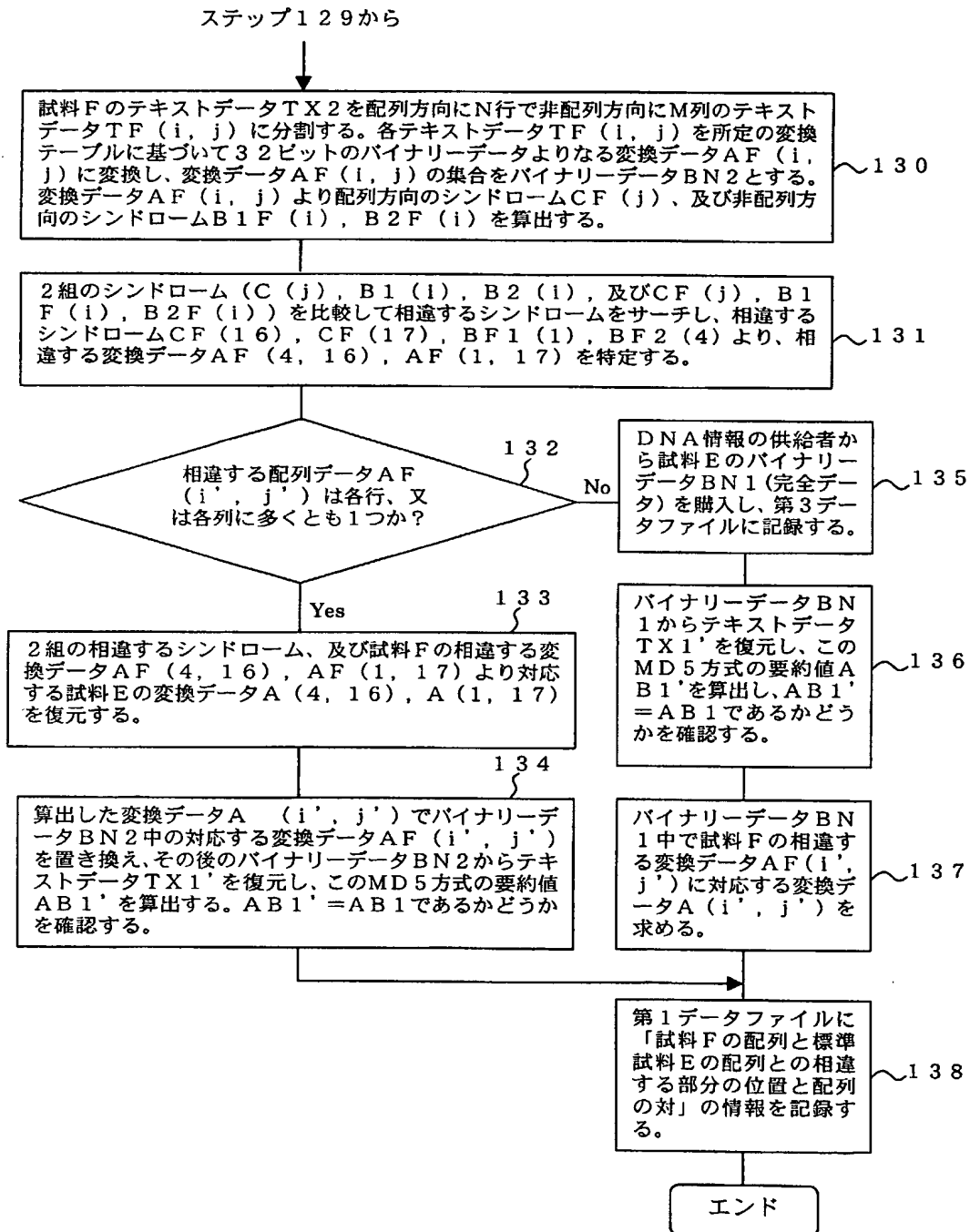
4/15

図 5



5/15

図 6



6/15



7

標準試料E				T(i, j)	
j	i	1	2	3	4
1	1	AGCTTTTCATTCTGAC	TGCAACGGGCAATAATG	TCTCTGTGTGGATTAA	AAAAAGAGTGTCTGAT
2	1	AGCAGCTTCTGAACTG	GTTACCTGCCCTGAGT	AAATTAATAATTTTATT	GACTTAGGTCACATAA
3	1	TACTTTAACCAATATA	GGCATAGCCACACAC	AGATAAAATTTACAGA	GTACACAACATCCATG
4	1	AAACGCATTAGCACCA	CCATTACCACACCAT	CACCATTTACCACAGGT	AACGGTGGGGGCTGAC
5	1	GGGTACAGGAACACAC	GAAAAAGCCCCGAC	TGACAGTGGGGCTTT	TTTTTTCGACCAAGG
6	1	TAACGAGGTAAACAAC	ATCGAGTGTGAAGT	TGCGCGGTACATCAGT	GGCAAAATGCAGAACGT
7	1	TTTCGCGTGTGCGG	ATATTCTGAAAGCAA	TGCCAGGCAGGGGCG	GTGCCCACCGTCTCT
8	1	CTGCCCCCGCCAAAT	CACCAACCACTGGTG	GGCATGATTGAAAAA	CCATTAGCGGCCAGGA
9	1	TGCTTTACCCCAATATC	AGCGATGCCGAACGTA	TTTTTCCGAACTTTT	GACGGACTCGCCGCC
10	1	GCCCAGCCGGGGTTCC	CGCTGGCCCAATTGAA	AACTTCGTCGATCAG	GAATTTGCCCAATAA
11	1	AACATGTCCTGCATGG	CATTAGTTTGTGGGG	CAGTCCCCGGATAGCA	TCAACGCTGGCTGAT
12	1	TTGCCGTGGCGAGAAA	ATGTCGATCGCCATTA	TGGCCGGCGTATTAGA	AGCGCGGGTCACAAC
13	1	GTTACTGTTATCGATC	CGGTCGAAAACTGCT	GGCAGTGGGCGATTAC	CTCGAATCTACCGTCG
14	1	ATATTGCTGAGTCCAC	CCGCCGTATTGGGCA	AGCCGCATTCGGCTG	ATCACATGGTGTGAT
15	1	GGCAGGTTTCACCGCC	GGTAATGAAAAAGGCG	AAC TGTTGTTGCTTG	ACGCAACGGTTCGGAC
16	1	TACTCTGCTGCGGTGC	TGGTGCCTGTTTACG	CGCCGATTGTTGCGAG	ATTTGGACGGACGTTG
17	1	ACGGGTCTATACCTG	CGACCCCGGTCAGGTG	CCCGATGCGGAGGTTGT	TGAAGTCGATGTCCTA
18	1	CCAGGAAGCGATGGAG	CTTTCCTACTTCGGCG	CTAAAGTTCTTCACCC	CCGCACCATTAACCCC
19	1	ATCGCCAGTTCAGAA	TCCCTTGCCCTGATTAA	AAATACCGGAAATCCT	CAAGCACAGGTACGC
20	1	TCAATGGTGCCAGCCG	TGATGAAGACGAAATTA	CCGGTCAAGGGCATTT	CCAATCTGAATAACAT
21	1	GGCAATGTTACAGGTT	TCTGGTCCGGGGATGA	AAGGATGGTCCGGCAT	GGCGCGCGGCTCTT
22	1	GCAGCGATGTCACGCG	CCCGTATTTCCGTGGT	GCTGATTACGCAATCA	TCTTCGGAATACAGCA
23	1	TCAGTTTCGCGTTCC	ACAAAGCGACTGTGTG	CGAGCTGAACGGGCAA	TGCAGGAAGAGTTCTA
24	1	CCTGGAAC TGAAAGAA	GGCTTACTGGAGCCGC	TGGCAGTGACGGAAACG	GCTGGCCATTATCTCG
25	1	GTGGTAGGTGATGGTA	TGCGCACCTTGCGTGG	GATCTCGCGGAAATTC	TTTGGCGCACTGGCCC
26	1	GCGCCAATATCAACAT	TGTCGCCATTGCTCAG	GGATCTTCTGAACGCT	CAATCTCTGCTGCTGT
27	1	AAATAACGATGATGCG	ACCACTGGCGTGGCGG	TTACTCATCAGATGCT	GTTCAATACCGATCAG
28	1	GTTATCGAAGTGTTG	TGATTGGCGTGGCGTG	CGTTGGCGGTGGCGTG	CTGGAGCAACTGAAGC
29	1	GTACGCAAGCTGGCT	GAAAGATAACATATC	GACTTACGTGTCTGCG	GTGTTGCCAACTCGAA
30	1	GGCTCTGCTCACCAAT	GTACATGGCTTTAATC	TGGAAAACTGGCAGGA	AGAACTGGCGCAAGCC
31	1	AAAGAGCCGTTTAAATC	TGCGGGCGCTTAATCG	CCCTGTTGAAAGAAATAT	CATCTGCTGAACCCGG
32	1	TCATTGTGACTGCAC	TTCCAGCCAGGCAGTG	GGGATCAATATATGCCG	ACTTCTCTGCGCGAAGG

7/15

8



j	標準試料E				A(i, j)	C(j)
	i	1	2	3		
1		lbfe3ed2	d82560cd	eeddd4f0	0011ded3	e3135362
2		1861b42d	7cada747	03c03fcf	4bc5e2c0	e4a37e03
3		cbf0a0cc	58c66212	13003c84	72208e8d	a9d7cdef
4		0263c628	a3ca28a3	8a3ca217	097656d2	39e0e7b4
5		67214088	4001a98a	d21d95bf	11f92805	7939a7d6
6		c245c20a	36477d07	e5972387	580d8427	3631e6bf
7		fed9df69	33ed4060	dal61561	75a29ebb	827fd3e5
8		b6aa6803	8a0a2b5d	64d3d000	a3c65a14	494ebd74
9		dbf2a0ce	1936909c	ffda42ff	4952e69a	3e565b03
10		6ala55fa	9b5983d0	0bf9e4e1	43f6a030	55645edb
11		08deb635	8f1fd155	876a5318	e09b66d3	00044f75
12		f69d6440	37939a3c	d69673c4	19997882	1e60eac2
13		7cb7ce4e	979002db	587558f2	b90eca79	25cbf494
14		33db47a2	a69cf658	1a63e96d	388d76d3	2d699e3a
15		585fe29a	5c340059	0b5d76f5	26097e92	e5fad87a
16		cb66d976	d6dadfc9	9a4f7d91	3f52527d	7c33894d
17		255eccad	92a6785d	a93645f7	d07937ac	31b4c2ad
18		a1419351	bfaceb59	b01f6e2a	a628f2aa	b737027e
19		39a87e84	eaf6b4f0	032940eb	818a1726	a9528b85
20		e3d76869	d341243c	a5e0563f	a0ed0c23	fde5e107
21		5837e19f	ed7a5534	054d7963	596667bf	a4661715
22		61937899	a9cfe9d7	6d3c9838	efa43218	68442cc0
23		elfed9fa	20192ddd	91b42560	d85047ec	6c1c7523
24		ad42d010	5bcb51a6	d61d2509	6d68f3b9	4c943a78
25		75c5d35c	d98af675	4ee5903e	fd62d6a	9bdc8779
26		66833823	de8f6e1	53bed09b	83bb79d7	1c667976
27		030934d9	28b59d99	f2e384db	7e0ca4e1	9cae1c2e
28		7ce41dfd	d3d67975	9f59766d	b5182d06	a52c3ae5
29		78601b5b	410c08ce	4bc9ded9	77da0b90	7d100e92
30		5bb6e283	7235af0e	d402d614	10b5981a	b2a41fbf
31		011a7f0e	e566f019	ae740433	8edb42a5	23d0b6df
32		e3df4b62	fala161d	65383369	2fad9905	72df2ded
B1(i)		935ab0e2	f4d95e21	1891405c	1616e995	
B2(i)		aafa481c	e846c00e	3558b73f	43d9f669	

図 9

標準試料 E					C (j)
i	1	2	3	4	
j					
1					e3135362
2					e4a37e03
3					a9d7cdef
4					39e0e7b4
5					7939a7d6
6					3631e6bf
7					827fd3e5
8					494ebd74
9					3e565b03
10					55645edb
11					00044f75
12					1e60eac2
13					25cbf494
14					2d699e3a
15					e5fad87a
16					7c33894d
17					31b4c2ad
18					b737027e
19					a9528b85
20					fde5ef07
21					a46617f5
22					68442cc0
23					6c1c7523
24					4c943a78
25					9bdc8779
26					1c667976
27					9caefc2e
28					a52c3ae5
29					7d100e92
30					b2a4ffbf
31					23d0b6df
32					72df2ded
B1 (i)	935ab0e2	14d95e21	1891405c	16f6e995	
B2 (i)	aafa481c	e846c00e	3558b73f	43d9f669	

j	i	1	試料F		2	3	TF (i, j)	
1			AGCTTTTCATCTCTGAC		TGCAACGGGGCAATATG	TCTCTGTGTGGATTAA	AAAAACAGTGTCTGAT	
2			AGCAGCTTCTCAACTG		GTTACCTGCCGTAGGT	AAATTAATAATTTTAT	GACTTAGGTCACTAAA	
3			TACTTTAACCAATATA		GGCATAGCGCACAGAT	AGATAAAAATTACAGA	GTACACAACTCCATG	
4			AAACGCATTAGCACCA		CCATTACCACACAGT	CACCATTAACACAGT	AACGGTCCGGGCTGAC	
5			GCGTACAGGAACACACA		GA AAAAGCCCCGACCC	TGACAGTCCGGGCTTT	TTTTTTCGACCAAGG	
6			TAACGAGGTAAACAACC		ATCGAGTGTGAAGT	TCGGCGGTACATCAGT	GGCAATGCAGAACGT	
7			TTTCGTGCTGTGGCG		ATATTCTGGAAGCAA	TGCCAGGCAGGGGCAG	GTGGCCACCGTCTCT	
8			TCGCCCTCCGCCAAAAT		CACCAACCACTTGGTG	GGGATGATTGA AAAA	CCATTAGCGGCCAGGA	
9			TGCTTTACCCCAATATC		AGCGATGCCGAACGTG	TTTTTCCGCAACTTT	GACGGCACTCGCGCC	
10			GCCCAAGCGGGGTTC		CGCTGGCGCAATTGAA	AACTTTCGTCGATCAG	GAATTTGCCCAATAA	
11			AACATGTCTCTGCATGG		CATTAGTTTGTGGGG	<u>CAGTCCCCGGATAGCA</u>	TCAACCGTCCGCTGAT	
12			TTGCCGTGGCGAGAAA		ATGTCGATCGCCATTAA	TGGCCGGCGATTAGA	AGCCGCTGGTCAACAAC	
13			GTTACTGTTATCGATC		CGGTGCAAAAACCTGCT	GGCAGTGGGGCAATTAC	CTCGAATCTACCGTGC	
14			ATATTGCTGAGTCCAC		CCGCCGTATTGCGGCA	AGCCGCATTCCGGCTG	ATCACATGGTCTGCTGAT	
15			GGCAGGTTTCAACGCCC		GGTAATGAAAAGGCG	AACTGGTGGTGTGG	ACGCAACGGTTCCGAC	
16			TACTCTGCTCGGGTGC		TGGCTGCCCTGTTTACG	CGCCGATTGTGGCGAG	ATTGGACATTATGGC	
17			GGCCAACTTATACCTG		CGACCCCGGTCAGGTG	CCCGATGCGAGGTTGT	TGAAGTCGATGTCTCTA	
18			TCAGGAACCGATGGAG		CTTTCTCTACTTCGGCG	CTAAAGTCTTTCACCC	CCGACCATTAACCCCC	
19			TATCGCCAGTTCAGGA		TCCCTTCCCTGATTAA	AAATACCGGAATCTCT	CARGCACCAGGTACGC	
20			TCAATGGTCCACGCC		TGAAGAAGACGAATTA	CCGGTCAAGGCATTT	CCAACTCTGAATAACAT	
21			GGCAATGTTACGGCTT		TCTGGTCCGGGGATGA	AAGGATGTCGGCAT	GGGGCGCGGCTCTTT	
22			GCAGCGATGTCACGGC		CCCGTATTCCGTGTT	CGTGATTACGCCATCA	TCTTCCGAATACAGCA	
23			TCAGTTTCTGCGTTCC		ACAAAGCGACTGTGTG	CGAGCTGAACGGGCAG	TGCAAGGAAGATTCTA	
24			CCTGGAACCTGAAAGAA		GGCTTACTGGAGCCGC	TGGCAGTGCAGGGAACG	GCTGGCCATTATCTCG	
25			GTGGTAGGTGATGGTA		TGCGCACTTTCGTGG	GATCTCGCGGAATTC	TTTCCGCACTGGCCC	
26			CGCCCAATATCAACAT		TGTGCGCATTGTCTCAG	GGATCTCTGAACGCT	CAATCTCTGTCTGGT	
27			AAATAACGATGATCGG		ACCATGGCGTCCGCG	TTACTCATCAGATGCT	GTTCAAATACCGATCAG	
28			GTTATCGAAGTGTITG		TCATTGGCTCGGTGG	GCTTGGCGGTGCGCTG	CTGGAGCAACTGAAC	
29			GTACGCAAGCTGGCT		GAAGAAATAACATATC	CGATTCTGTCTGGC	GTGTTGCCAACTCGAA	
30			GGCTCTGCTCACCAAT		GTACATGGCCTTAATC	TGGAACACTTGCAGGA	AGAAGTGGCGGCAAGC	
31			AAAGAGCCGTTTATC		TCGGCGCTTAATTCC	CCCTGTGAAAGAAATAT	CATCTGCTGAACCCGG	
32			TCAITGTGACTGCAC		TTCAGCCAGGCAGTG	CGGGATCAATATGCCG	ACTTCTCGCGCAAGG	

10/15

111

i	1	試料 F	2	3	AF (i, j)	4	CF (j)
1	1bfe3ed2		d82560cd	eedd4f0	0011ded3		e3135362
2	186fb42d		7cada747	03c03fcf	4bc5e2c0		e4a37e03
3	cbf0a0cc		58c66212	13003c84	72208e8d		a9d7cdef
4	0263c628		a3ca28a3	8a3ca217	097656d2		39e0e7b4
5	67214088		4001a98a	d21d95bf	ff92805		7939a7d6
6	c245c20a		36477d07	e5972387	580d8427		3631e6bf
7	fed9df69		33ed4060	da161561	75a29ebb		827fd3e5
8	b6aa6803		8a0a2b5d	64d3d000	a3c65a14		494ebd74
9	dbf2a0ce		1936909c	ffda42ff	4952e69a		3e565b03
10	6ala55fa		9b5983d0	0b19e4e1	43f6a030		55645edb
11	08deb635		8f11df55	876a5318	e09b66d3		00044f75
12	f69d6440		37939a3c	d69673c4	19997882		1e60eac2
13	7cb7ce4e		979002db	587558f2	b90eca79		25cbf494
14	33db47a2		a69cf658	1a63e96d	388d76d3		2d699e3a
15	585fe29a		5c340059	0b5d76f5	26097e92		e5fad87a
16	cb66d976		d6dadfc9	9a4f7d91	3f533cd6		7c3373a6
17	5a0bccad	AF (1, 17)	92a6785d	a93645f7	d07937ac		6661c2ad
18	a1419351		bfacbe59	b01fbe2a	a628f2aa		b737077e
19	39a87e84		eat6b4f0	032940eb	818a1726		a9528b85
20	e3d76869		d341243c	a5e0563f	a0ed0c23		fde5ef07
21	5837e19f		ed7a5534	054d7963	59b667bf		a46617f5
22	61937899		a9cfe9d7	6d3c9838	ef4a3218		68442cc0
23	elfed9fa		20192ddd	91b42560	d85047ec		6c1c7523
24	ad42d010		5bcb51a6	d61d2509	6d68f3b9		4c943a78
25	75c5d35c		d98af675	4ee5903e	fda62d6a		9bdc8779
26	66833823		de68f6e1	53bed09b	83bb79d7		1c667976
27	030934d9		28b59d99	f2e384db	7e0ca4e1		9cacfc2e
28	7ce41dfd		d3d67975	9f59766d	b5182d06		a52c3ae5
29	78601b5b		410c08ce	4bc9ded9	77da0b90		7d100e92
30	5bb6e283		7235af0e	d402d614	10b5981a		b2a4ffbf
31	011a7f0e		e566f0f9	ae740433	8edb42a5		23d0b6df
32	e3df4b62		fala161d	65383369	2fad9905		72df2ded
B1F (i)	c807b0e2		f4d95e21	1891405c	f6f6e995		
B2F (i)	aafa481c		e846c00e	3558b73f	43d9e0c2		

1 2

j	i	1	2	3	4	CF (j)
1						e3135362
2						e4a37e03
3						a9d7cdef
4						39e0e7b4
5						7939a7d6
6						3631e6bf
7						827fd3e5
8						494ebd74
9						3e565b03
10						55645edb
11						00044f75
12						1e60eac2
13						25cbf494
14						2d699e3a
15						e5fad87a
16						7c3373a6
17						6661c2ad
18						b737027e
19						a9528b85
20						fde5ef07
21						a46617f5
22						68442cc0
23						6c1c7523
24						4c943a78
25						9bdc8779
26						1c667976
27						9caefc2e
28						a52c3ae5
29						7d100e92
30						b2a4ffb1
31						23d0b6df
32						72df2ded
B1F (i)		c807b0e2	f4d95e21	1891405c	16f6e995	
B2F (i)		aa1a481c	e846c00e	3558b73f	43d9e0c2	

12/15

図 1 3

試料 G										
	i	1	2	3	4	5	6	7	8	
1		MRVL	KFGG	TSVA	NAER	FLRV	ADIL	ESNA	RQGQ	
2		VATV	LSAP	AKIT	NHLV	AMIE	KTIS	GQDA	LPNI	
3		SDAE	RIFA	ELLT	GLAA	AQPG	FPLA	QLKT	FVDQ	
4		EFAQ	IKHV	LHGI	SLLG	QCPD	SINA	ALIC	RGEK	
5		MSIA	IMAG	VLEA	RGHN	VTVI	DPVE	KLLA	VGHY	
6		LEST	VDIA	ESTR	RIAA	SRIP	ADHM	VLMA	GFTA	
7		GNEK	GELV	VLGR	NGSD	YSAA	VLAA	CLRA	DCCE	
8		IWTD	VDGV	YTCD	PRQV	PDAR	LLKS	MSYQ	EAME	
9		LSYF	GAKV	LHPR	TITP	IAQF	QIPC	LIKX	TGNP	51
10		QAPG	TLIG	ASRD	EDEL	PVKG	ISNL	NNMA	MFSV	
11		SGPG	MKGM	VGMA	ARVF	AAMS	RARI	SVVL	ITQS	52
12		SSEY	SISF	CVPQ	SDCV	RAER	AMQE	EFYL	ELKE	
13		GLLE	PLAV	TERL	AIIS	VVGD	GMRT	LRGI	SAKF	53
14		FAAL	ARAN	INIV	AIAQ	GSSE	RSIS	VVVN	NDDA	
15		TTGV	RVTH	QMLF	NTDQ	VIEV	FVIG	VGGV	GGAL	
16		LEQL	KRQQ	SWLK	NKHI	DLRV	CGVA	NSKA	LLTN	→
17		VHGL	NLEN	WQEE	LAQA	KEPF	NLGR	LIRL	VKEY	
18		HLLN	PVIV	DCTS	SQAV	ADQY	ADFL	REGF	HVVT	
19		PNKK	ANTS	SMDY	YHQL	RYAA	EKSR	RKFL	YDTN	54
20		VGAG	LPVI	ENLQ	NLLN	AGDE	LMKF	SGIL	SGSL	
21		SYIF	GKLD	EGMS	FSEA	TTLA	REMG	YTEP	DPRD	
22		DLSG	MDVA	RKLL	ILAR	ETGR	ELEL	ADIE	IEPV	
23		LPAE	FNAE	GDVA	AFMA	NLSQ	LDDL	FAAR	VAKA	
24		RDEG	KVLR	YVGN	IDED	GVCN	VKIA	EVDG	NDPL	
25		FKVK	NGEN	ALAF	YSHY	YQPL	PLVL	RGYQ	AGND	
26		VTAA	GVFA	DLLR	TLSW	KLGV	0	0	0	

図 1 4

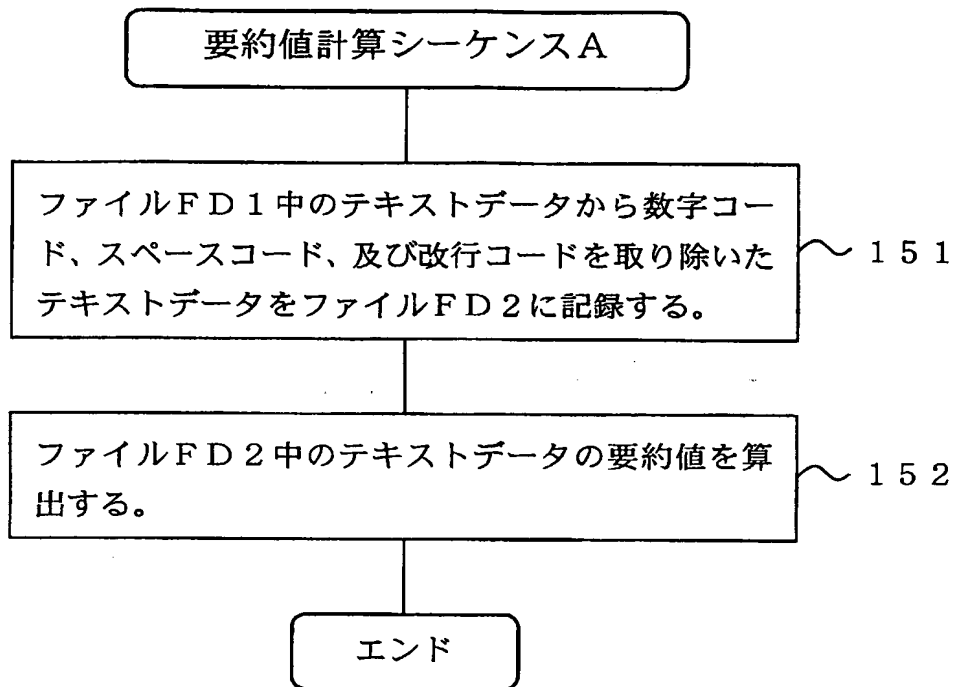
15	TTGV	RVTH	QMLF	NTDQ	VIEV	FVIG	VGGV	GGAL	51
16	LEQL	KRQQ	SWLK	NKHI	DLRV	CGVA	NSKA	LLTN	52
17	VHGL	NLEN	WQEE	LAQA	KEPF	NLGR	LIRL	VKEY	55

56B

56A

13/15

図 1 5



14/15

図 1 6

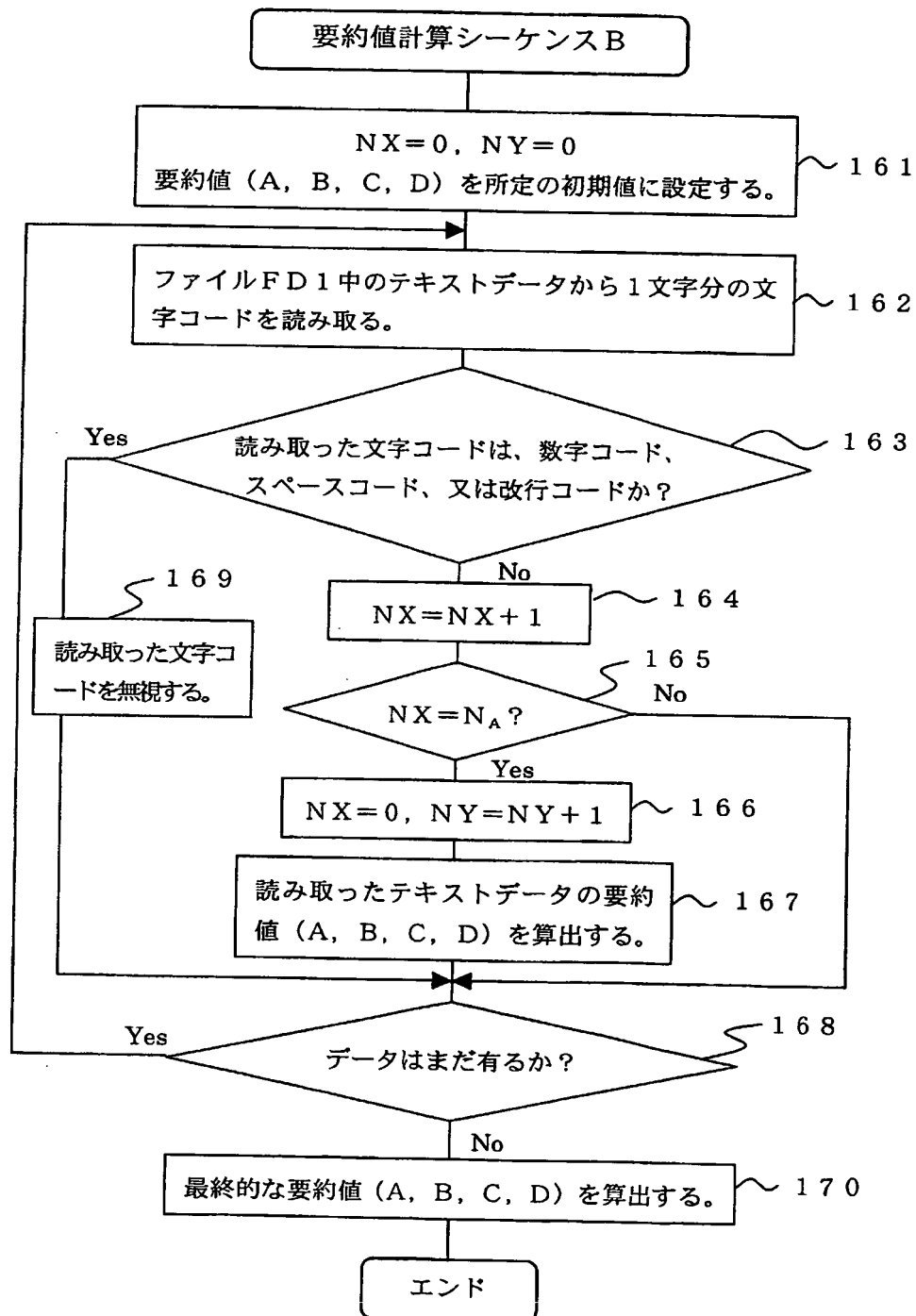
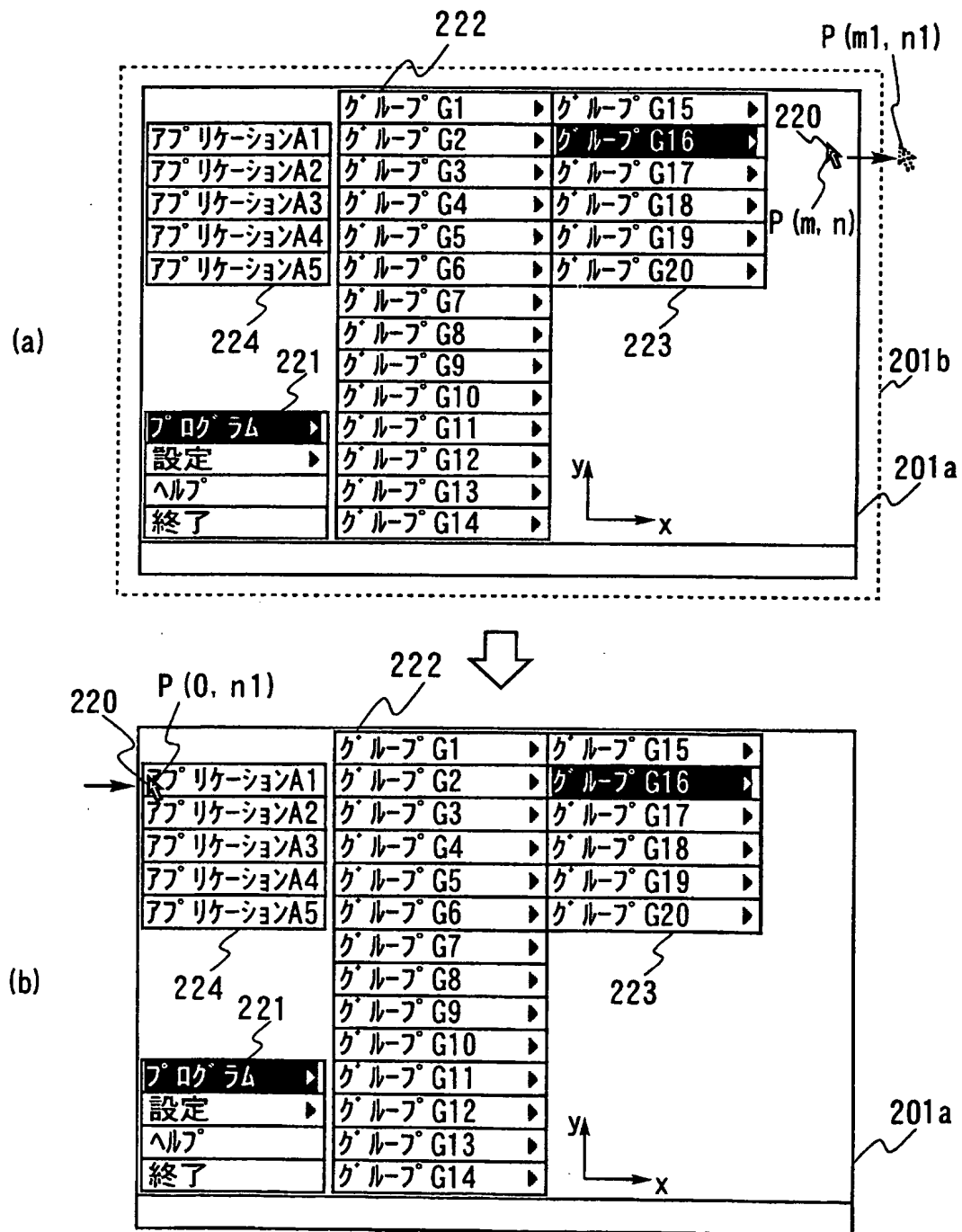


図 1 7



SEQUENCE LISTING

5 <110> Omori, Satoshi

<120> Method and apparatus for recording information of nucleotide
sequence and amino acid sequence

10 <130> 2001F01

<140>

<141>

15 <150> JP 2000-117343

<151> 2000-04-19

<150> JP 2000-149122

<151> 2000-05-19

20

<160> 3

<170> PatentIn Ver. 2.0

25 <210> 1

<211> 2048

<212> DNA

<213> Escherichia coli

<400> 1

5 agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60
tgatagcagc tttctgaactg gttacctgcc gtgagtaaata taaaatttta ttgacttagg 120
tcactaaata ctttaaccaa tataggcata gcgcacagac agataaaaaat tacagagtac 180
acaacatcca tgaaacgcat tagcaccacc attaccacca ccatcaccat taccacaggt 240
aacggtgcgg gctgacgct acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300
10 cttttttttt cgaccaaagg taacgaggta acaacatgc gagtgttgaa gttcggcggg 360
acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420
aggcaggggc aggtggccac cgtcctctct gccccgccca aaatcaccaa ccacctggtg 480
gcgatgattg aaaaaacat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540
cgtatttttg ccgaactttt gacgggactc gccgccgcc agccgggggt cccgctggcg 600
15 caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660
agtttggttg ggcagtggcc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720
atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgttata 780
gatccggtcg aaaaactgct ggcagtgggg cattacctcg aatctaccgt cgatattgct 840
gagtccacc gccgtattgc ggcaagccgc attccggctg atcacatggt gctgatggca 900
20 ggtttcaccg ccggtaatga aaaaggcgaa ctggtggtgc ttggacgcaa cggttccgac 960
tactctgctg cgggtgctggc tgctgttta cgcgccgatt gttgcgagat ttggacggac 1020
gttgacgggg tctatactg cgaccgcgt cagggtcccc atgcgaggtt gttgaagtcg 1080
atgtcctacc aggaagcgat ggagctttcc tacttcggcg cttaaagtct tcacccccgc 1140
accattacc ccatcgccca gtccagatc ctttgctga ttaaaaatac cggaaatcct 1200
25 caagcaccag gtacgtcat tggtgccagc cgtgatgaag acgaattacc ggtcaagggc 1260
atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320

gtcggcatgg cggcgcgcggt ctttgcagcg atgtcacgcg cccgtatttc cgtggtgctg 1380
 attacgcaat catcttccga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440
 cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg ctactggag 1500
 ccgctggcag tgacggaacg gctggccatt atctcggtgg taggtgatgg tatgcgccacc 1560
 5 ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620
 gccattgctc agggatcttc tgaacgctca atctctgtcg tggtaaataa cgatgatgcg 1680
 accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgatacagg tategaagtg 1740
 tttgtgattg gcgtcgggtg cgttggcggg gcgctgctgg agcaactgaa gcgtcagcaa 1800
 agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctgaaggct 1860
 10 ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920
 aaagagccgt ttaatctcgg gcgcttaatt cgctcgtga aagaataica tctgctgaac 1980
 ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
 cgcaagg 2048

15 <210> 2
 <211> 2048
 <212> DNA
 <213> Escherichia coli

20 <400> 2
 agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60
 tgatagcagc ttctgaactg gttacctgcc gtgagtaaata taaaatttta ttgacttagg 120
 tcaactaaata ctttaaccaa tataggcata gcgcacagac agataaaaaat tacagagtac 180
 acaacatcca tgaaacgcat tagcaccacc attaccacca ccataccat taccacaggt 240
 25 aacgggtgcgg gctgacgcgt acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300
 cttttttttt cgaccaaagg taacgaggta acaaccatgc gagtggtgaa gttcggcggg 360

acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420
aggcaggggc aggtggccac cgctctctct gccccgccca aaatcaccaa ccacctgggtg 480
gcgatgattg aaaaaacat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540
cgtatttttg ccgaactttt gacgggactc gccgccgcc agccgggggtt cccgctggcg 600
5 caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660
agtttggttg ggcagtgcc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720
atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgtttatc 780
gatccggtcg aaaaactgct ggcagtgggg cattacctcg aatctaccgt cgatattgct 840
gagtcaccc gccgtattgc ggcaagccgc attccggctg atcacatggt gctgatggca 900
10 ggtttcaccc ccgtaatga aaaaggcgaa ctggtggtgc ttggacgcaa cggttccgac 960
tactctgctg cggctgtggc tgctgttta cgcgccgatt gttgcgagat ttggacatta 1020
tggcggccaa cttatactg cgaccgcgt cagggtcccc atgcgaggtt gttgaagtcg 1080
atgtcctacc aggaagcgat ggagctttcc tacttcggcg ctaaagtctc tcacccccgc 1140
accattacc ccacgcca gttccagatc cttgcctga ttaaaaatac cggaaatcct 1200
15 caagcaccag gtacgtcat tggtgccagc cgtgatgaag acgaattacc ggtcaagggc 1260
atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320
gtcggcatgg cggcgcgcgt cttgcagcg atgtcacgcg cccgtatttc cgtggtgctg 1380
attacgcaat catcttcga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440
cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg cttactggag 1500
20 ccgctggcag tgacggaacg gctggccatt atctcgggtg taggtgatgg tatgcgcacc 1560
ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620
gccattgctc aggatcttc tgaacgcica atctctgtcg tggtaaataa cgatgatgcg 1680
accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgatcaggt tatcgaagtg 1740
tttgtgattg gcgtcgggtg cgttggcggt gcgctgctgg agcaactgaa gcgtcagcaa 1800
25 agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctcgaaggct 1860
ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920

aaagagccgt ttaatctcgg gcgcttaatt cgcctcgtga aagaatatca tctgctgaac 1980
 ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
 cgccaagg 2048

5 <210> 3
 <211> 820
 <212> PRT
 <213> Escherichia coli

10 <400> 3
 Met Arg Val Leu Lys Phe Gly Gly Thr Ser Val Ala Asn Ala Glu Arg
 1 5 10 15

Phe Leu Arg Val Ala Asp Ile Leu Glu Ser Asn Ala Arg Gln Gly Gln
 15 20 25 30

Val Ala Thr Val Leu Ser Ala Pro Ala Lys Ile Thr Asn His Leu Val
 35 40 45

20 Ala Met Ile Glu Lys Thr Ile Ser Gly Gln Asp Ala Leu Pro Asn Ile
 50 55 60

Ser Asp Ala Glu Arg Ile Phe Ala Glu Leu Leu Thr Gly Leu Ala Ala
 65 70 75 80

25 Ala Gln Pro Gly Phe Pro Leu Ala Gln Leu Lys Thr Phe Val Asp Gln

85 90 95

Glu Phe Ala Gln Ile Lys His Val Leu His Gly Ile Ser Leu Leu Gly
100 105 110

5

Gln Cys Pro Asp Ser Ile Asn Ala Ala Leu Ile Cys Arg Gly Glu Lys
115 120 125

Met Ser Ile Ala Ile Met Ala Gly Val Leu Glu Ala Arg Gly His Asn
10 130 135 140

Val Thr Val Ile Asp Pro Val Glu Lys Leu Leu Ala Val Gly His Tyr
145 150 155 160

15

Leu Glu Ser Thr Val Asp Ile Ala Glu Ser Thr Arg Arg Ile Ala Ala
165 170 175

Ser Arg Ile Pro Ala Asp His Met Val Leu Met Ala Gly Phe Thr Ala
180 185 190

20

Gly Asn Glu Lys Gly Glu Leu Val Val Leu Gly Arg Asn Gly Ser Asp
195 200 205

Tyr Ser Ala Ala Val Leu Ala Ala Cys Leu Arg Ala Asp Cys Cys Glu
25 210 215 220

Ile Trp Thr Asp Val Asp Gly Val Tyr Thr Cys Asp Pro Arg Gln Val
225 230 235 240

Pro Asp Ala Arg Leu Leu Lys Ser Met Ser Tyr Gln Glu Ala Met Glu
5 245 250 255

Leu Ser Tyr Phe Gly Ala Lys Val Leu His Pro Arg Thr Ile Thr Pro
260 265 270

10 Ile Ala Gln Phe Gln Ile Pro Cys Leu Ile Lys Asn Thr Gly Asn Pro
275 280 285

Gln Ala Pro Gly Thr Leu Ile Gly Ala Ser Arg Asp Glu Asp Glu Leu
290 295 300

15 Pro Val Lys Gly Ile Ser Asn Leu Asn Asn Met Ala Met Phe Ser Val
305 310 315 320

Ser Gly Pro Gly Met Lys Gly Met Val Gly Met Ala Ala Arg Val Phe
20 325 330 335

Ala Ala Met Ser Arg Ala Arg Ile Ser Val Val Leu Ile Thr Gln Ser
340 345 350

25 Ser Ser Glu Tyr Ser Ile Ser Phe Cys Val Pro Gln Ser Asp Cys Val
355 360 365

Arg Ala Glu Arg Ala Met Gln Glu Glu Phe Tyr Leu Glu Leu Lys Glu
370 375 380

5 Gly Leu Leu Glu Pro Leu Ala Val Thr Glu Arg Leu Ala Ile Ile Ser
385 390 395 400

Val Val Gly Asp Gly Met Arg Thr Leu Arg Gly Ile Ser Ala Lys Phe
405 410 415

10

Phe Ala Ala Leu Ala Arg Ala Asn Ile Asn Ile Val Ala Ile Ala Gln
420 425 430

15

Gly Ser Ser Glu Arg Ser Ile Ser Val Val Val Asn Asn Asp Asp Ala
435 440 445

Thr Thr Gly Val Arg Val Thr His Gln Met Leu Phe Asn Thr Asp Gln
450 455 460

20

Val Ile Glu Val Phe Val Ile Gly Val Gly Gly Val Gly Gly Ala Leu
465 470 475 480

Leu Glu Gln Leu Lys Arg Gln Gln Ser Trp Leu Lys Asn Lys His Ile
485 490 495

25

Asp Leu Arg Val Cys Gly Val Ala Asn Ser Lys Ala Leu Leu Thr Asn

500 505 510

Val His Gly Leu Asn Leu Glu Asn Trp Gln Glu Glu Leu Ala Gln Ala
515 520 525

5

Lys Glu Pro Phe Asn Leu Gly Arg Leu Ile Arg Leu Val Lys Glu Tyr
530 535 540

10

His Leu Leu Asn Pro Val Ile Val Asp Cys Thr Ser Ser Gln Ala Val
545 550 555 560

Ala Asp Gln Tyr Ala Asp Phe Leu Arg Glu Gly Phe His Val Val Thr
565 570 575

15

Pro Asn Lys Lys Ala Asn Thr Ser Ser Met Asp Tyr Tyr His Gln Leu
580 585 590

Arg Tyr Ala Ala Glu Lys Ser Arg Arg Lys Phe Leu Tyr Asp Thr Asn
595 600 605

20

Val Gly Ala Gly Leu Pro Val Ile Glu Asn Leu Gln Asn Leu Leu Asn
610 615 620

Ala Gly Asp Glu Leu Met Lys Phe Ser Gly Ile Leu Ser Gly Ser Leu
25 625 630 635 640

Ser Tyr Ile Phe Gly Lys Leu Asp Glu Gly Met Ser Phe Ser Glu Ala
645 650 655

Thr Thr Leu Ala Arg Glu Met Gly Tyr Thr Glu Pro Asp Pro Arg Asp
5 660 665 670

Asp Leu Ser Gly Met Asp Val Ala Arg Lys Leu Leu Ile Leu Ala Arg
675 680 685

10 Glu Thr Gly Arg Glu Leu Glu Leu Ala Asp Ile Glu Ile Glu Pro Val
690 695 700

Leu Pro Ala Glu Phe Asn Ala Glu Gly Asp Val Ala Ala Phe Met Ala
705 710 715 720

15 Asn Leu Ser Gln Leu Asp Asp Leu Phe Ala Ala Arg Val Ala Lys Ala
725 730 735

Arg Asp Glu Gly Lys Val Leu Arg Tyr Val Gly Asn Ile Asp Glu Asp
20 740 745 750

Gly Val Cys Arg Val Lys Ile Ala Glu Val Asp Gly Asn Asp Pro Leu
755 760 765

25 Phe Lys Val Lys Asn Gly Glu Asn Ala Leu Ala Phe Tyr Ser His Tyr
770 775 780

Tyr Gln Pro Leu Pro Leu Val Leu Arg Gly Tyr Gly Ala Gly Asn Asp
785 790 795 800

5 Val Thr Ala Ala Gly Val Phe Ala Asp Leu Leu Arg Thr Leu Ser Trp
805 810 815

Lys Leu Gly Val
820

10

15

20

25

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/03324

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl.⁷ H03M7/30, G06F19/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl.⁷ H03M7/30, G06F19/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho (Y1,Y2) 1926-1996 Toroku Jitsuyo Shinan Koho (U) 1994-2001
Kokai Jitsuyo Shinan Koho (U) 1971-2001 Jitsuyo Shinan Toroku Koho (Y2) 1996-2001

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 10-151125 A (Corp Miyuki K.K.), 09 June, 1998 (09.06.98), Fig. 5 (Family: none)	1~5, 12, 15, 21, 22, 28, 33, 34
A	JP 10-272123 A (Hitachi, Ltd.), 13 October, 1998 (13.10.98), Fig. 1 (Family: none)	6~11, 13, 14, 16~20, 23~27, 29~31, 35~39

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
18 July, 2001 (18.07.01)

Date of mailing of the international search report
31 July, 2001 (31.07.01)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))
Int. Cl⁷ H03M7/30, G06F19/00

B. 調査を行った分野
調査を行った最小限資料 (国際特許分類 (IPC))
Int. Cl⁷ H03M7/30, G06F19/00

最小限資料以外の資料で調査を行った分野に含まれるもの
日本国実用新案公報 (Y1, Y2) 1926-1996年
日本国公開実用新案公報 (U) 1971-2001年
日本国登録実用新案公報 (U) 1994-2001年
日本国実用新案登録公報 (Y2) 1996-2001年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	JP 10-151125 A (株式会社コーポレーションミューキ)、9. 6月. 1998 (09. 06. 98)、図5 (ファミリーなし)	1~5, 12, 15, 21, 22, 28, 33, 34
A	JP 10-272123 A (株式会社日立製作所)、13. 10月. 1998 (13. 10. 98) 図1 (ファミリーなし)	6~11, 13, 14, 16~20, 23~27, 29~31, 35~39

☐ C欄の続きにも文献が列挙されている。

☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
「O」 口頭による開示、使用、展示等に言及する文献
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
「&」 同一パテントファミリー文献

国際調査を完了した日
18. 07. 01

国際調査報告の発送日
31.07.01

国際調査機関の名称及びあて先
日本国特許庁 (ISA/JP)
郵便番号100-8915
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)
石井 研一

5K 8124

電話番号 03-3581-1101 内線 3555